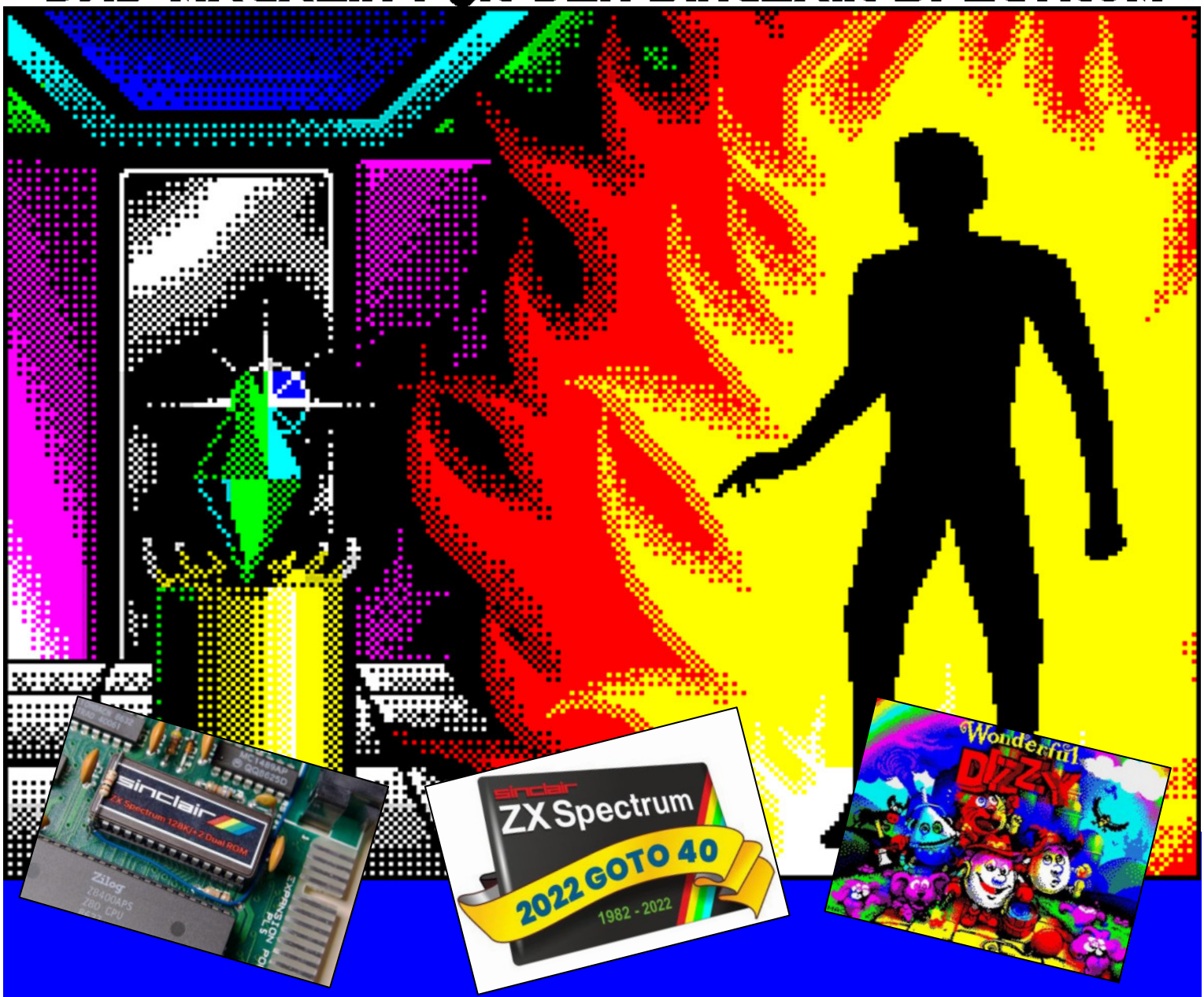
SUC-SESSION

DAS MAGAZIN FÜR DEN SINCLAIR SPECTRUM



- News
- Dual-ROM für den Spectrum +2
- Spiele-Hack »*Monjas*«
- Adventure-Lösung »*Calling*«
- ZX SPECTRUM48 Systemvariable
- Computer-Klassiker *Sharp MZ700 & MZ800*
- Und vieles mehr …

DIE

231.

AUSGABE IHRES MAGAZINS

FÜR ABONNENTEN MIT SCENE+ DISKETTE

DIE NEUESTEN UND BESTEN PROGRAMME GLEICH MITGELIEFERT

VORWORT

Hallo liebe SUC-Leser,

es sind schon wieder fast vier Monate vergangen, seit die letzte SUC-Session erschienen ist. Viele Dinge sind seitdem geschehen. Natürlich beschäftigt uns alle der unselige Konflikt Russlands mit der Ukraine – die Nachrichten sind voll davon. Es schein, als wolle ein Autokrat seine Großmachtsträume ausleben. Was mich persönlich daran ärgert, ist dass es wieder ein Autokrat geschafft hat, einen Eisernen Vorhang quer durch Europa zu ziehen – nur diesmal weiter östlich. Als vor 33 Jahren besagter Eiserner Vorhang fiel, glaubten alle, der permanente Konflikt zwischen Ost und West sei vorbei. Aber wir haben uns offenbar getäuscht. Die Lehren aus dem 2. Weltkrieg sind offenbar vergessen. Helmut Schmidt, unser Alt-Bundeskanzler hat irgendwann gesagt, »lieber 100 Stunden erfolglos verhandeln als einen einzigen Schuss abfeuern.« Dem kann ich mich nur anschließen und hoffen, dass das alles bald vorbei ist.

Zum Glück scheint sich wenigstens die Corona-Pandemie dem Ende entgegen zu neigen. Die aktuelle Virus-Variante ist zwar hoch ansteckend, verursacht aber offenbar einen weniger gefährlichen Verlauf; zumindest laut den Medien. Das tröstet diejenigen aber nicht, die Angehörige verloren haben oder an Long Covid leiden.

Schluss mit dem Genöle!

Freuen wir uns auf das aktuelle Heft, in dem **FRANK DOPIERALA** erklärt, wie man ein Doppel-Eprom in einen Spectrum +2 einbaut, **THOMAS** die ersten Geheimnisse seiner Hacker-Karriere ausplaudert und uns **HARALD LACK** mit der Lösung des Spiels **»CALLING«** vertraut macht.

Außerdem steht die ***SPECTRUM40*** vor der Tür. **THOMAS EBERLE** und **LEE FOGARTY** laden zur ***SPECTRUM40 EVENT*** am **Samstag, den 30. April** nach Walsall ins *Bescot (Bank's)Stadium ein*. Auf der Karte, die auf der Website zu finden ist, erkennt man auch die amerikanische Botschaft – ungefähr vier Meilen Luftlinie entfernt findet sich ein McDonald's…

Alle Infos findet ihr auf der englischsprachigen Website. Der Link auf die Website lautet:

`https://spectrumevents.website/`

Zum Schluss noch ein Aufruf!

Wir, die Redaktion der **SUC-SESSION,** suchen immer noch jemanden, männlich, weiblich oder gemischt, der/die/das sich berufen fühlt, die Spielerezensionen des **SPECTRUM NEXT** zu übernehmen. Bitte meldet euch zahlreich – Thomas zahlt euch auch beim nächsten Treffen eine Currywurst!

Spaß beiseite! Uns geht die Zeit aus. Aus diesem Grund findet sich in diesem Heft auch kein Artikel zum Next-Basic. Wir können nicht alles selber schreiben. Traurig aber wahr: Es gibt noch andere Sachen im Leben als den **ZX SPECTRUM** - auch wenn das kaum zu glauben ist.

Wie dem auch sei…

Wir wünschen euch allen viel Spaß mit dem Heft.

Viele Grüße,

Joachim und Thomas

22. März 2022

INHALT

Vorwort, Inhalt, Impressum 2
Leserbriefe 3
New in Kürze 6
SCENE+ Diskette No. 76 7
Vom Artikel zum Heft 7
Wie hacke ich ein Spiel: »**MONJAS**« 11
Systemvariablen **ZXS 48** 13
Nachtrag zu den Heft 229 & 230 23
Dual-ROM beim **SPECTRUM +2** 24
Sharp **MZ 700** und **MZ 800** 25
Adventurelösung **»CALLING«** 29
SPECTRUM NEXT Spiele **»GOLD COMPILATION«** 32
Games 2019/2022 33
SPECTRUM40 in Walsall/England 40

IMPRESSUM

Das **SUC-Session**-Magazin erscheint in Deutsch und in Englisch und kann beim **SpectrumUserClub** zusammen mit dem Disk Magazin **SCENE+** abonniert werden.
Infos hierzu bei Thomas Eberle.

Redaktion SUC-Session 232:
Joachim Geupel & Thomas Eberle

Redakteur SCENE+: Thomas Eberle

Clubleitung:
Mirko Seidel, Adresse: Birkenweg 8
89197 Weidenstetten
email: mirko.seidel1@gmail.com

Mitarbeiter an diese Ausgabe sind:

Joachim Geupel	Joa.Geu@gmx.de
Thomas Eberle	thomas.eberle@sintech-shop.de
Ellvis	
Harald Lack	
Frank Dopierala	

Die Rechte der Artikel sowie der Bilder und Grafiken liegen bei den einzelnen Autoren und Fotografen.

Wir danken allen Beteiligten für ihre Beiträge!

Redaktionsschluss für SUC SESSION 232

Mittwoch, 29. Juni 2022

LESERBRIEFE

Reinhard Hock **aus Trossingen hat leider einen defekten Spectrum +2A.**

Leider habe ich seit kurzem ein Problem mit meinem Spectrum +2 (A) 128k, vielleicht haben Sie einen Rat für mich:

Ich habe ein Divide-Modul an gestöpselt und erhielt dann ein merkwürdiges Bild, siehe Bild.

Inzwischen habe ich das Divide-Modul wieder entfernt. Wenn ich den Spectrum einschalte tut sich jedoch nichts mehr.

Woran kann das liegen bzw. wie könnte sich das beheben lassen?

Viele Grüße, Reinhard Hock

Hallo Herr Hock,

ich bin mir nicht sicher, aber viele Module, auch einige DIVIDE sind nicht für den Spectrum +2A gemacht, da dieser eine andere Busport-Belegung hat als die Vorgänger-Spectrums. Diese falsche Belegung kann zu einem Defekt des Computers führen. Auch möglich ist natürlich, dass das Interface grundsätzlich für den +2A geeignet ist, aber z.B. etwas schief aufgesteckt war oder beim Spectrum im laufenden Betrieb aufgesteckt wurde. Alle diese Dinge können zu einem Kurzschluss führen und da der Z80 direkt mit dem Busport verbunden ist, geht dabei sehr häufig der Prozessor kaputt, in wenigen aber schlimmen fällen kann auch die ULA betroffen sein. ULA wäre beim +2A so gut wie ein Totalschaden, der Z80 wiederum ist leicht auszutauschen. Bei SINTECH können Sie auch einen Z80 ordern. Der Z80 sollte gesockelt sein und ist leicht auszutauschen, ich denke einen Versuch ist es an der Stelle wert.

Reinhard Hock aus Trossingen hat sich nochmal gemeldet mit Fragen zur Hardware.

Hallo Herr Eberle,

anbei ein Foto meines Z80. Können Sie daraus erkennen, ob der Chip gesockelt ist?

Ich schaue mich gerade nach einem Ersatz-Spectrum um und bin sowohl an einem 48K-Model dran als auch an einem weiteren +2 128K. Meinem Divide-Modul traue ich allerdings nicht mehr so recht über den Weg und bin deshalb am Umschauen nach einer möglichen Alternative.

In Ihrem Online-Shop ist mir diesbezüglich das DivMMC-Interface aufgefallen. Dies entspricht ja in etwa der Divide bzw. ist sogar noch eine Spur komfortabler. Dazu habe ich folgende Fragen:

- *Ist es möglich mittels des von mir bestellten ZX-EXT sowohl den VGA-/Joystick-Adapter als auch ein DivMMC-Modul an den Spectrum anzuschließen und somit beide zusammen zu nutzen?*
- *Laufen mittels des DivMMC auf einem 48K-Spectrum dann auch 128K-Spiele? (Vermutlich nicht…)*

Vielen Dank im Voraus für Ihre Antworten,

Reinhard Hock

Hallo nochmal, Herr Hock.

Der Z80 Chip auf dem Bild ist gesockelt, also nicht fest eingelötet. Er kann vorsichtig herausgehoben und dann ein neuer Z80 eingesetzt werden. Man muss dabei darauf achten, dass auch alle Füßchen in der Fassung sind. Auf genau die gleiche Richtung achten.

Egal ob Divide oder DivMMC, beides läuft mit ZX-EXT und ZX-VGA zusammen. Das DivMMC hat den Vorteil, dass es auf jeden Fall auch für den +2A ist, bei ihrem Divide hatte ich ja bedenken, das könnte auch die Ursache für den Crash sein.

Um 128k Spiele zu spielen bedarf es auch 128k Speicher, ein Divide oder DivMMC erweitert den Arbeitsspeicher des Spectrum nicht. Auch hat ein 48k Spectrum keinen AY-Soundchip. Hierfür haben wir zwar ein Interface, aber irgendwann sind es auch genug Interfaces.

Ein drittes Mal in dieser Ausgabe Reinhard Hock aus Trossingen, diesmal mit einem Tastaturproblem:

Hallo Herr Eberle,

habe mir die Tage einen Spectrum 48 k+ angeschafft, der mit der »richtigen« Tastatur, ein echt schöner Computer. :-)

Interessant ist, dass die Tastatur nur dann komplett funktioniert, wenn ich das Gerät aufschraube und das Oberteil mit der Tastatur etwas vom Gehäuseunterteil anhebe. Setze ich das Oberteil dann wieder in die richtige Position zurück, funktionieren ein paar Tasten nicht mehr. Ich vermute mal, dass das an der transparenten Membrane liegt, welche die Tastatur mit dem Motherboard verbindet?

Falls ja, sollte es nicht so schwierig sein diese auszuwechseln? Erfreulich: Mein Divide-Modul funktioniert mit dem Spectrum + einwandfrei. :-)

Viele Grüße,

Reinhard Hock

Hallo Herr Hock,

ja, die Membrane hat wohl einen feinen Riss, der sich in einer bestimmten Position der Membrane schließt und in Normalposition die Verbindung unterbricht. Ein Wechsel der Membrane dürfte das Problem lösen.

Der Einbau einer Membrane ist nicht ganz einfach, die Folie besteht aus mehreren Lagen die zusammengepresst werden müssen. Daher müssen die Schrauben in genau der richtigen Stärke angezogen werden, dies herauszufinden ist ein bisschen ein Suchspiel. Aber mit etwas ausprobieren kriegt man das hin.

Dass das Divide jetzt geht beweist, dass mit dem anderen Spectrum etwas nicht stimmt.

Paul Veltjens **aus Geilenkirchen meldete sich per e-mail zur geplanten Einstellung der SCENE+:**

Ich habe ja keine Diskettenlaufwerke und brauche somit keine.

Schon aus sentimentalen Gründen fände ich es aber schade wenn es keine Diskversion mehr gäbe.

Mein Vorschlag: verknappe die Anzahl und mache entweder teurer oder vergib sie gegen Höchstgebot. Das macht es eventuell interessanter.

Liebe Grüße Paul

Hallo Paul,

es geht nicht um das Kopieren von Disketten, sondern um das einmalige erstellen der Ausgabe. Alle Kassettenladebefehle müssen geändert werden auf Diskette. Manchmal muss man sogar MC Laderoutinen umschreiben oder gar den ganzen Code verschieben wegen zu hohem RAMTOP. Dazu wird das Ganze noch speicher-technisch optimiert, durch Packprogramme werden die Programme mind. ein Drittel kürzer und es geht mehr auf eine Diskette. Solche Sorgen hat man mit SD Karten nicht, die arbeiten mit Tape-Laderoutinen und haben kein Speicherplatzproblem (zumindest keines wo es auf 15 kB ankommt). Für die Scene+ verwende ich sehr viel Zeit und Energie, die ich einfach optimal einsetzen will. Mir tut es auch leid, aber ich wage zu behaupten, dass die allermeisten unser Magazin nicht wegen der Diskette abonnieren. Für ein Produkt, das kaum gefragt ist dann soviel Energie reinzustecken, ist einfach nicht sinnvoll. Ich lasse mich gerne eines Besseren belehren, aber so stellt sich die Situation für mich momentan dar.

Lothar Ebelshäuser **aus Kall-Scheuen würde ein einstellen der SCENE+ sehr bedauern:**

Hallo Thomas, auch mal wieder was von mir.

Der Clubbeitrag ist überwiesen worden. Natürlich möchte ich auch weiterhin dabei sein. Das Info ist eingetroffen, und wieder wirklich gut geworden. Alle Artikel sind gut gemacht und leicht zu verstehen. Jetzt freue ich mich auf die nächste Diskette. Und das ist auch schon mein Hauptanliegen.

Du schriebst so etwas zuletzt. Im Laufe von so vielen Jahren haben sich natürlich sehr viele Disketten von der Scene+ angesammelt. Immer wieder hole ich die eine oder andere Diskette raus und spiele einige Zeit.

Darauf verzichten würde mir sehr, sehr schwer fallen. Würden es die Scene+ auch auf SD Karte für den Spectrum Next geben, wäre es wieder ein Grund, sich mehr mit dem Rechner zu befassen.

Aber beides wäre wahrscheinlich wirtschaftlich zu aufwendig. Von der vielen Arbeit abgesehen. Also mach bitte weiter mit den Disketten, die sind ja immer besser geworden. Und ich freue mich jedes Mal wie jeck auf die nächste.

Mit TAP-Files kenne ich mich nicht so aus.

Viele Grüße aus der Eifel, Lothar

Hallo Lothar, danke für deine Meldung. Es ist natürlich schön wenn man liest, dass die erstellte SCENE+ auch genutzt wird, ehrlich gesagt habe ich aber das Gefühl, dass viele Menschen die SCENE+ eben nur haben, weil die mit dem Magazin geliefert wird. Aber in erster Linie geht es den meisten um das Heft, vor allem Disketten nutzt kaum noch jemand. Natürlich könnten wir die SCENE+ auch auf SD-Karte veröffentlichen, nur sehe ich darin keinen Sinn. Auf der SD-Karte wäre nur ein TAP-File. Jeder der ein Interface für SD-Karte oder einen Spectrum NEXT etc. hat, hat auch eine SD Karte und in der Regel haben die 4GB. Darauf passen nicht nur alle SCENE+ Ausgaben, die jemals erstellt wurde, sondern generell die gesamte Spectrum-Softwarebibliothek. Anstatt also viele SD Karten zu nutzen, könnte ein User einfach das TAP-File auf seine schon vorhandene SD-Karte kopieren, er könnte Unterordner anlegen und hätte es schön sortiert, ohne die SD Karte wechseln zu müssen. Bei dieser Überlegung fällt dann auf, dass die SCENE+ als TAP-File dann generell unnötig ist. Die Diskettenversionen werden gepackt, um Speicherplatz zu sparen und der Ladesyntax angepasst. Beides ist bei TAP unnötig, Speicherplatz ist wie gesagt genug da und der Syntax von Kassette bleibt erhalten.

Meine Idee ist stattdessen, eine allen zugängliche Club-Website mit den Programmlinks der Programme, die wir empfehlen. So gibt es die SCENE+ sozusagen zum selber zusammenstellen, wer will kann sich dann nur auf Spiele oder nur auf Demos konzentrieren. Die Frage ist, gibt es noch User die ein +D/Opus haben und kein Interface für SD–Karten? Ich bezweifle es. Sollte es noch jemand geben, würden die etwas auf der Strecke bleiben, zwar kann man via Emulator auch die TAP-Files laden und auf Diskette kopieren, aber man bräuchte dazu wohl mind. einen WIN2000 Rechner mit einem Diskettencontroller. Und wer hat so was noch…außer mir? Die Frage ist also: Wie viele User BRAUCHEN die SCENE+ wirklich, weil sie gar keine andere Möglichkeit haben, Software zu überspielen. Und dann die Frage, wenn es nur ein bis zwei User sind, können die nicht doch auf ein DivMMC wechseln als modernerer, sicherer und letztlich auch einfacherer Speicher? Gerne lese ich dazu.

Auch ***Norbert Opitz*** **aus Wittenberg hat einen Vorschlag zur Fortsetzung der SCENE+:**

Hallo Thomas !

Fertige das Disk-Magazin Scene+ als TAP-File und die SUC-Session als PDF-File, damit du weniger Arbeit für diese Informations-Mittel hast.

Tschüss, Norbert

Hallo Norbert,

danke für deine Idee. Wichtig ist zu wissen, dass weder Joachim, noch Mirko oder ich das ganze hier für Geld tun, die Beiträge sind in erster Linie dazu da, die Kosten zu decken. Auch PDF und TAP kostet einen Beitrag, aber auch diese Beiträge dienen zur Kostendeckung, wenn die Kosten auch vielleicht an anderer Stelle entstehen. Die Magazine nur noch digital zu vertreiben, würde fast alle Kosten einsparen und wir könnten das Heft gratis verteilen. Allerdings frage ich mich, ob ein Gratis-Heft dann für die Leser noch so einen Wert hätte. Das Internet ist voll von gratis Informationen, aber wer nutzt die? Wir denken, dass es User gibt, die selbst ein ausgedrucktes Heft in der Hand haben wollen und dafür eben mehr bezahlen. Jedem steht es aber frei durch Mitarbeit das Heft günstiger oder gar gratis zu bekommen. Dies ist unser Konzept, damit sichergestellt ist, dass Mitarbeit

belohnt wird und wir auch wirklich Hilfe bekommen. Den alleine können Joachim und ich das Heft nicht füllen, wir hätten dann nur noch ein 8 Seiten Informationsblatt. Was die SCENE+ angeht, sind schon alle Programme die wir veröffentlichen als TAP erhältlich, die Links liefern wir ja in unseren Spiele- und Demo-Reviews. Eine reine TAP-Ausgabe scheint mir überflüssig, die TAP fiel an als digitale MB02 Version, aber wenn es keine MB02 – Version mehr gibt, wird es auch kein TAP mehr geben.

Harald Lack aus Raubling meldete sich in einer anderen Sache, die uns überaus peinlich war, die wir aber nicht verschweigen möchten:

Hallo Thomas,
hab Grad von Joachim erfahren, dass mir die letzten drei Hefte fehlen. Du hattest doch bisher immer eine Mail geschickt, wenn das neue Heft draußen war? Hast Du die richtige Mailadresse? Joachim schickt mir die drei Hefte. Das nur zu Deiner Info! Gruß Harald

Hallo Harald,
tatsächlich habe ich unseren fleißigsten Leser vergessen. Tatsächlich ist es so: Ein Abo erhält man durch Bezahlung. Für seine Artikel erhält man Punkte, die man zur Bezahlung verwenden kann. Das Abo verlängert sich aber nicht automatisch, du musst quasi proaktiv das Abo verlängern und zur Bezahlung deine Punkte einsetzen.
So ist es generell vorgesehen. In deinem Fall habe ich aber wohl dir auch gar keine Erinnerung geschickt, du bist ja eh in jedem Heft vertreten, da ist es sinnfrei eine Liste zu machen. Aber in der Theorie gibt es diese Liste eben und so ging es schief. Dies nur zur Erklärung , wie das passieren konnte. Joachim hat dir die fehlenden Hefte geschickt, Entschuldigung noch von meiner Seite.

Josef Prokas aus Sibirna/Tschechien verlängerte sein Abo:

Hallo Thomas,
ich möchte mein Abo erneuern. Ich hatte schon vorab einen Betrag gesendet für die ganzen vorherigen Hefte der letzten Jahrzehnte, die ich bekommen durfte. Vielen Dank dafür.
*TAP ist für mich perfekt, denn ich habe keinen anderen älteren Massenspeicher. Ich nutze nur ESX-Dos (mit **MB03+** und **eLeMeNt ZX**) und konzentriere mich hauptsächlich auf meine Webseite `https://zxfiles.samcoupe.cz` (ich habe auch die Seite `https://sam.speccy.cz` erneuert und muss nun esxdos.samcoupe.cz ebenfalls updaten.*
*Ich denke, nach all den Jahren ist es nun Zeit, dass man die ganzen bekannt gewordenen Speccy-Erweiterungen in ein Gerät integriert. **LMN128** hat nun den **Sega Controller** und **Pentagon-4096** in seine **eLeMeNt/MB** hardware eingebunden.*

`https://oldcomp-cz.translate.goog/viewtopic.php?f=129&t=8898&start=45&_x_tr_sl=cs&_x_tr_tl=de&_x_tr_hl=cs&_x_tr_pto=wapp#p136560`

*Ich arbeite immer noch an neuen Versionen der Anleitungen für das **eLeMeNt ZX**, **LnxCopy**, **LnxCommander**... und technischen Dokumentationen.*

*Wie meine Kontakte mit russischen Freunden mir sagen, sind auch immer mehr Personen aus Russland interessiert an den westlichen Speccy-Erweiterungen. **LMN128** hat bereits **TurboSound**, **GeneralSound**, **SounDrive**, **Pentagon 1024** und **Z-Controller** implementiert. Und jetzt produzieren die russischen Spectrum-User **Pentagons** mit **DivMMC**. Speziell der **Sizif-512** scheint wirklich toll zu sein:*

`https://sam.speccy.cz/zxtoday.html`

*Das **esxDOS** mit seiner eingebauten **TRDOS** und Kassettenemulation ist perfekt, um diverse Software umzusetzen.*
*Der zweite Typ meiner Aktivitäten ist das Konvertieren von alten Anwendungen und das **esxDOS**, z.B. **Datalog** oder **Text Machine**. Ja, ich benutze dieses Programm immer noch auf der Arbeit :-)*

`https://sam.speccy.cz/datalog.html`

Ich bin mir nicht sicher, ob es hier noch größeren Bedarf für diese Aktivitäten gibt. Wenn dem so ist, lasst es mich wissen und ich könnte einen Artikel für das Magazin schreiben.
Wie auch immer, nach meiner Erfahrung stecken die Menschen in ihrer Vergangenheit fest und es dauert viele, viele Jahre und Monate, um auf andere, bessere Erweiterungen zu wechseln.
*Und **SpecNext**, natürlich, es brachte meiner Meinung nach die Menschen zu einer anderen, Amiga-ähnlichen Plattform...*
Grüße, Josef

Vielen Dank Josef für den Einblick in Deine umfangreichen Tätigkeiten. Tatsächlich gab es für den SAM Coupe mal eine kleine aber feine Gemeinde auch in Deutschland, ich denke auch, dass einige unserer Leser noch einen SAM haben. Selbst empfand ich den SAM aber auch nicht wirklich als Spectrum-Nachfolger, da nicht einmal das Bank-Switching gleich funktioniert wie beim Spectrum und damit Spectrum 128k Programme nicht kompatibel sind. Lediglich der 48k Modus kann emuliert werden. Der Spectrum NEXT hat einen etwas anderen Ansatz, zumindest funktionieren mal alle Spectrum Programme und der Spectrum NEXT kann auch wie ein Spectrum genutzt werden. Auch in punkto Design ist er ein würdiger Nachfolger. Das der NEXT eben aber auch noch mehr bietet, was teilweise dem Amiga ähnelt, ist richtig, meiner Meinung nach wird das aber nur von wenigen genutzt. Tatsächlich ist aber eins gleich geblieben: Der NEXT ist, ähnlich wie der SAM, einfach in Basic zu programmieren und das unterscheidet beide Geräte sehr erheblich vom Amiga. Insofern sind beide Geräte als Nachfolger des Spectrums aus der Sicht zu sehen, dass die Bedienung recht einfach zu erlernen ist. Der Spectrum NEXT eben in einer etwas neueren Version und mit einem eigenen Standard.
Tatsächlich ist es aber so, das sehr viel Unkenntnis im westlichen Europa und der Welt herrscht über das, was in der Tschechischen Republik oder auch Russland entwickelt wird. Deine Webseite gibt da einen Einblick, aber ich denke ich spreche für alle wenn ich sage, dass wir gern mehr davon hören, vor allem auch die Kernfrage: Wo kann ich diese tollen Sachen bekommen... Daher hören wir gern mehr von dir.

Leserbriefe wie immer an
thomas.eberle@sintech-shop.de

NEWS IN KÜRZE

NEUE SOFTWARE BEI SINTECH

SINTECH in Deutschland hat nun eine Rubrik für neue Spectrum Spiele. Dazu hat **SINTECH** den Vertrieb der Spiele von **SEBASTIAN BRAUNERT** übernommen, derzeit sind dies »MORITZ THE STRIKER« und »MORITZ ON THE AUTOBAHN«. **SINTECH** bemüht sich auch um den Vertrieb weiterer Titel.

Auch ist die Ausweitung des Vertriebs bei **SINTECH.UK** und **SINTECH.CZ** geplant. Die Titel im deutschen Shop findet man hier :

https://www.sintech-shop.de/retro-atari-commodore-sinclair-etc/sinclair/spectrum/neue-software

Aufgrund der speziellen Brexit-Regeln versendet **SINTECH-DEUTSCHLAND** nicht nach UK und aufgrund der Kriegssituation auch nicht in die Ukraine oder Russland.

SPECTRUM NEXT OS »NEXTDOWS«

Ein interessantes Betriebssystem für den **NEXT** ist momentan in der Entwicklung. Nach dem booten

findet man sich dann in einer **WINDOWS**-ähnlichen (oder sagen wir lieber **MAC OS** , das klingt besser) Oberfläche mit Symbolen und Ordnern, wie wir es von besagten System her kennen. Inklusive ist auch ein File Explorer zum Kopieren, löschen und ausführen von Dateien. Manch einer ist eine solche Oberfläche suspekt, ich kann aber alle beruhigen, denen es nicht genug *»Retro«* ist. den das Beta-Diskinterface kannte eine solche Oberfläche (Vision) schon 1984. Im Vergleich dazu ist **»NEXTDOWS«** wesentlich umfangreicher, bietet auch z.B. Netzwerk-Einstellungen, Soundeinstellung Bildschirmeinstellung, Uhrzeit... Das Ganze ist noch in der Beta-Phase mit Version 0.7.1 . Mehr erfahren und testen kann nur, wer das Projekt finanziell unterstützt (der Betrag kann dabei frei gewählt werden, auch 1 Euro ist möglich). Die Unterstützung anbieten kann man auf folgender Seite:

http://zx.duefectucorp.com/2022/02/19/support-us/

Wir werden sicher über dieses Projekt noch ausführlicher informieren, wenn eine komplette Version erschienen ist.

WEITER VERZÖGERUNG BEIM SPECTRUM NEXT KS2

Die weltweite Chip Knappheit geht auch an uns Retro-Usern nicht vorbei. Beim **SPECTRUM NEXT** bedeutet das eine weitere Verzögerung, da die programmierbaren FPGA Chips »**SPARTAN-6**« nicht mehr am Markt verfügbar sind. Das Team musste das ganze Board um designen, damit es jetzt mit dem leistungsfähigeren (aber auch teureren) **ARTIX-7**–Chip läuft, welcher offenbar am Markt besser verfügbar ist. Momentan hat das Team also nun zwei Designs fertig und hofft darauf, für ein Design den richtigen Chip in ausreichender Menge bekommen zu können.

Leider wird dadurch für die Spectrum Next aus dieser Produktion eine andere Firmware benötigt werden, dies ist nicht zu ändern.

FILMEMACHER ANDY REMIC VERSTORBEN

Der Autor von Filmen wie **»MEMOIRS OF A SPECTRUM ADDICT«** hatte auch einen Film über die Entstehung des **SPECTRUM NEXT** geplant. Dazu wird es wohl nicht mehr kommen, zumindest nicht von Andy. Er verstarb am 26. Februar aus uns unbekannten Gründen im Alter von nur 51 Jahren. Rest in piece, Andy.

ZOSYA VERÖFFENTLICHT SCREENSHOTS

Aufgrund der Kriegssituation hat **ZOSYA** vorerst den Vertrieb von **SPECTRUM SOFTWARE** einstellen müssen. Dies hat das Team nicht gehindert, weiter zu programmieren und das Spiel »**RUBINHO CUCACARACHA**« soll demnächst online veröffentlicht werden. Bis dahin vertreibt uns das Team die Zeit und veröffentlicht Screenshots von ca. 100 weiteren Projekten, die viel tolle Grafik und Action versprechen. Die Posts von **NATASHA ZOTOVA** findet man am besten auf Facebook:

https://www.facebook.com/groups/164156683632183/search/?q=natasha%20zotova

SCENE+ DISKETTE NO. 76

VON THOMAS EBERLE

Wer rechtzeitig abonniert hatte, hatte vor dieser Ausgabe auch schon die Diskette Nr. 76 bekommen (oder TAP). Wieder waren viele tolle Programme darauf:

- OXYGEN – Szene-Demo
- KARATEKA – Szene-Demo
- 70908 – Szene-Demo
- VALLEY OF RAINS – Actionspiel
- CELL3326 – Labyrinthspiel
- BLOCK DUDE – Knobelspiel
- The HOUSE ON THE OTHER SIDE of THE STORM – Text-Adventure
- JUST A GAL – Rennspiel
- YOYO´S GREAT ADVENTURE – Action-Adventure
- VOID – Szene-Demo
- SPACE INVADERS – Arcade-Ballerspiel
- COLORISTIC – Knobelspiel
- MIRE MARE – Labyrinthspiel
- MONJAS – Action-Adventure
- THE HUMANS – Knobelspiel
- AGENT BLUE – Platform / Actionspiel

Wir danken in erster Linie den Autoren, in erster Linie auch ZOSYA Entertainment für die Erlaubnis auch die eigentlich kommerziell vertriebenen Spiele nutzen zu dürfen. Insgesamt eine bunte Auswahl hauptsächlich von Spielen aller Genres, gemixt mit den besten Demos. Leider diesmal keine Sound- oder Grafikshows und wie so häufig leider auch keine Utilities (die sind rar). Trotzdem eine gelungene Ausgabe, deren Umsetzung auch nicht so einfach war.

Dies ist leider auch der Grund für die folgenden Zeilen: Die Umsetzung von Spielen auf Diskette ist schwierig und zeitraubend. Niemals ist nur der Loader zu ändern. Oft werden auch bei PD-Programmen die Loader geschützt, in fast allen Fällen sind die Programme nicht speicheroptimiert und müssen noch gepackt werden damit mehr auf eine Diskette passt. Das größte Problem ist aber natürlich die Umsetzung auf +D und dann auch auf Opus. Beide Floppy-Interfaces belegen etwas Speicher im Hauptspeicher des Spectrums und die wenigsten Programmierer nehmen Rücksicht darauf. Optimimiert wird nur für Kassette oder vielleicht für TR-DOS (Russland), aber nicht für Opus und +D. Die Arbeit macht mir aber nichts aus, sondern die Userzahlen. Für die Opus haben wir noch 4 Abonnenten, wenig mehr für +D. Alle anderen wählen das TAP-Format. Auf TAP-Format muss man nicht viel ändern an den Programmen, das ist sehr einfach zu machen. Der Vorschlag war, dass wir einfach nur noch das TAP-Format machen. Aber dies halte ich für sinnfrei, die Programme sind doch alle im Internet im TAP-Format erhältlich, warum sollte ich mir die Arbeit machen diese zusammenzufassen, wo doch Ellvis in jeder SUC-Session Ausgabe Spiele und Demos beschreibt samt Download-Link.

Da die Abos noch laufen, wird es noch eine Ausgabe der SCENE+ geben. Mit der Ausgabe 77 werden wir dieses Magazin dann aber einstellen und uns ganz auf die SUC-Session konzentrieren.

Gleichwohl möchte ich die Diskussion um Alternativen eröffnen. Wie wäre es mit einer Webseite um Disk-Konvertierungen herunterzuladen? Wer hat Interesse? Gerne höre ich von euch.
Schreibt uns gerne per Brief oder email.
te

VOM ARTIKEL ZUM HEFT

EIN PERSÖNLICHER RÜCKBLICK

VON JOACHIM GEUPEL

Die SUC-Session entsteht seit etwas mehr als drei Jahren auf meinem Computer. Ich habe die Anzahl der Beiträge nicht gezählt, die ich beigesteuert habe, es sind aber eine ganze Menge. Im Allgemeinen sind es mindestens zwei, meistens sind es mehr.

Dazu gekommen bin ich ausversehen und weil ich ein guter Mensch bin – zumindest meistens. Und das geschah so:

Auf der Rückfahrt von der ersten Spectrumania, an der ich teilgenommen habe, haben Mirko und Thomas versucht, mich davon zu überzeugen, die SUC-Session zu übernehmen, was von mir kategorisch abgelehnt wurde. Ich habe ihnen leichtsinni-

gerweise erzählt, dass ich schon mehrfach die Clubzeitschrift für zwei Clubs bzw. Vereine herausgegeben habe. Das erste Mal war es zwischen 1985 und 1989. Damals waren die privaten Mailboxen und Datenfernübertragung mittels Akustikkoppler und irgendwann einem höchst illegalen Hayes-Modem angesagt. Als echte deutsche Vereinsmeier haben wir, die User einer Mailbox namens ZEROPAGE einen Verein gegründet, der die Datenfernübertragung populär machen wollte. Was ja auch geklappt hat. Das zweite und das dritte Mal war ich einem SF-Club angeschlossen, der sich mit einer Serie der 60er-Jahre befasste, etwas genauer, nämlich REN DHARK[1]. Hier habe ich zum Schluss sogar die Vervielfältigung und den Versand übernommen. Der Aufwand war immens und hat tief in mein Privatleben eingegriffen. Es hat irgendwann keinen Spaß mehr gemacht. Ich hatte viel Arbeit und wurde nur kritisiert. Als der Punkt erreicht war, an dem der Ärger den Spaß überwogen hat, habe ich mein Engagement eingestellt. Zum Schluss habe ich den Club sogar zu Grabe getragen. Das wird hier im Spectrum User Club bisher zum Glück nicht der Fall sein und Spaß mach es immer noch.

Thomas hat es dann doch irgendwann geschafft, mich davon zu überzeugen, das Heft zu übernehmen. Ich hab mich lange gestäubt! Ehrlich! Aber jetzt fahre ich einen Ferrari, der von Thomas finanziert wird! Echt wahr!

Natürlich nicht! Thomas hat so viel Arbeit am Hals, dass er mit dem Heft nicht mehr nachgekommen ist. Und weil Thomas in der Zwischenzeit zu einem guten und vertrauenswürdigen Freund geworden ist, wollte ich ihm helfen. Ok, allerdings ein Baseball-Schläger ist ein überzeugendes Argument…

Meine Zielsetzung war und ist es, das Heft regelmäßig heraus zu geben und es lesbar zu gestalten. Ich glaube, das ist mir gelungen – zumindest hat sich bisher keiner beschwert. Thomas und ich arbeiten ausgezeichnet zusammen. Er sieht die Fehler, die ich nicht sehe, weil ich viel zu nah am Heft bin. Seine eigenen Fehler sieht man erst, wenn das Heft gedruckt ist… Wenn ich im Vorwort oder in einem Artikel zu direkt bin oder zu böse rüberkomme, entschärft er sie. Ich beneide ihn oft genug um seine Ruhe.

Thomas! Betrachte dich als gelobt!

ARTIKEL SCHREIBEN

Ich habe eigentlich nie daran gedacht, über die Grammatik und die Orthographie der deutschen Sprache zu schreiben. Ich komme aber offenbar nicht darum herum.

Ich habe zwei Anliegen an euch!

Natürlich! Schreibt Artikel! Sonst wird das Heft zu dünn. Unser Ziel ist es, zischen 36 und 44 Seiten zu machen. Dazu benötigen wir Beiträge. Und zugegebenermaßen geht mir das Material aus. Mir fällt bald nichts mehr Neues ein, zumindest nichts Spectrum-spezifisches. Dass ich die Computer meiner Sammlung vorstelle, weil die Zeit zwischen 1980 und 1986 sehr interessant war, habt ihr sicher selbst festgestellt. Ich recherchiere zu den entsprechenden Computern, steuere - sofern vorhanden - persönliche Erlebnisse bei und stelle sie Euch vor. Aber irgendwann habe ich alle meine interessanten Geräte vorgestellt und diese Artikelserie wird beendet. Also: schreibt Artikel!

Das zweite Anliegen sind tatsächlich die Artikel, die ich bekomme.

Die meisten Artikel sind in Ordnung. Es ist eine Freude, sie zu lesen und es freut mich, dass es nach wie vor Menschen gibt, die sich ebenfalls mit einem Computer befassen, dessen Sturm-und-Drang-Zeit eigentlich Anfang der 80er war. Ich sitze hier an der exponiertesten Position, weil ich die Artikel, die von euch kommen, als erster lesen darf. Genau hier beginnt für mich die Arbeit.

Ich benötige zwischen drei und sieben Tage, um ein Heft zu erstellen. Je nachdem, was ich sonst noch zu tun habe, verbringe ich zwischen drei und sieben Stunden am Computer. Ich erstelle das Heft übrigens mit Word 2010, was eigentlich denkbar schlecht dazu geeignet ist. Word ist ok, es ist aber kein DTP-Programm. Aber ich kenne mich damit so gut aus, dass am Schluss das Heft dabei rauskommt, das ihr in den Händen haltet.

Mein Problem besteht darin, dass ich mehr Zeit brauche, die Artikel zu überarbeiten, weil zum Teil viele Rechtschreibfehler drin sind, oder ich Texte bekomme, deren Formatierung ich nicht übernehmen kann, als das Heft selbst zu erstellen. Die orthografischen Fehler sehe ich normalerweise auf den ersten Blick; ich korrigiere sei – fertig! Tatsächlich sind Texte mit den eigenwilligen Formatierungen für mich das eigentliche Problem. Im Folgenden erkläre ich, wie ich die Artikel gerne hätte.

Fangen wir von vorne an.

WAS ICH GERNE HÄTTE:

GANZ WICHTIG! ÜBERSCHRIFTEN!

Jeder Artikel benötigt eine Überschrift. Die Überschrift gibt einen ersten Anhaltspunkt, um was es in dem Artikel geht. Fehlt diese, kann ich – oder die anderen Leser – nicht abschätzen, um was es in dem Artikel geht. Es kann eine Spielebeschreibung sein oder die Vorstellung neuer Hardware, vielleicht ist es auch ein Bericht über ein Treffen oder die persönlichen Erlebnisse eines Lesers, der sich vorstellt. Ich kann das vorher nicht wissen. Demnach lese ich den Artikel durch und gebe ihm eine eigene Überschrift, die, wenn ich Pech habe, das Thema verfehlt und dann den Unwillen des Autors nach sich zieht. War alles schon da – wenn auch nicht im Spectrum User Club. Also: bitte eine Überschrift über den Artikel.

AUCH NOCH WICHTIG: VERFASSER DES ARTIKELS ANGEBEN

Ich bekomme immer wieder Artikel, die hochinteressant sind, ich jedoch keine Ahnung habe, von wem sie sind. Ich freue mich grundsätzlich über

[1] Das Copyright von Ren Dhark liegt beim HJB-Verlag

jeden Artikel, allerdings sollte ich auch wissen, wer ihn verfasst hat. Der Vorname oder nur die Namens-Initialen langen nicht.

Selbstverständlich nehme ich Rücksicht darauf, wenn der jeweilige Autor seinen Namen nicht genannt haben will; er wird dann auch nicht genannt, ich sollte aber zumindest intern wissen, von wem dieser Artikel ist, damit ich im gegebenen Fall rückfragen kann, wenn Unklarheiten aufgetaucht sind.

ABSÄTZE!

Ein Artikel sollte in Absätze unterteilt sein. Damit meine ich nicht verschiedene Abschnitte, die im Artikel ein neues Thema einleiten. Ich meine damit, dass wenn ein Gedanke, der in dem Absatz behandelt wird und einen Themenbereich umfasst, zu Ende gedacht ist, das Thema aber noch nicht beendet ist, muss ein Absatz eingefügt werden.

Beispiel aus Heft 230, Clive Sinclair, Abschnitt *»Die Jahre danach«:*

> *(…) In dieser Pokerrunde schied er als Erster aus, gewann aber 2003 die erste Staffel von Celebrity Poker Club und konnte 25.000 Pfund mit nachhause nehmen.*

Sir Clive Sinclair erhielt mehrere Ehrungen für seinen Beitrag zur Etablierung der Personalcomputer-Industrie im England. (…)

Hier werden die Jahre nach *Sinclair Research Ltd.* behandelt und von einem Lebensabschnitt in einen anderen übergeleitet. Also wurde beim Schreiben ein Gedankengang beendet und ein weiterer aufgenommen. Absätze erleichtern das Lesen und machen den Text verständlicher.

EIN SCHWIERIGES THEMA: NUMMERIERUNGEN UND AUFZÄHLUNGSZEICHEN!

Die unterschiedlichsten Aufzählungen der verschiedenen Artikel machen mir das Leben wirklich schwer.

Beispiel:

**** ……. Text***

(hier steht irgendein erklärender Text. Weil ich aber niemanden bloßstellen will, kommen jetzt zwei Zeilen Geschwafel von mir, gefolgt von den Buchstaben, wenn ich mich einmal quer über die Tastatur lehne: +äüpolörgnmfklawäplsfgml .ökjhtgvöcsoaöml)

**** …… Text***

(Diese Zeilen sind nur Fülltext ohne Sinn. Geht also in Deckung! Du auch, Thomas!)

(…)

Diese Aufzählungen muss ich alle von Hand entfernen, um sie durch die entsprechende Funktion aus Word zu ersetzen. Ebenso muss ich dann die Einrückungen ersetzen.

Word bietet die Möglichkeit, Aufzählungen zu automatisieren. Damit das Schriftbild einigermaßen gleichmäßig ist, verwende ich diese Formatierungsmöglichkeiten, seien es jetzt Nummerierungen oder nur Aufzählungszeichen.

Was ich mir wünschen würde, ist, dass, sollten Nummerierungen notwendig sein, am Anfang der Zeile nur eine Zahl steht, z.b. *»1.« oder »1.)«,* ohne dass irgendwelche Textzeichen wie *Punke, +, *,* etc. eingefügt werden. Hier bietet sich ein Tabulator an. Der wird mittels der Tab-Taste (⇨| (→|) eingefügt und hat einen festgelegten Abstand. Alle mir bekannten Textprogramme, selbst der Editor von Windows, kann Tab-Stopps setzen. Lehrzeichen, Sterne und der Gleichen sind unnötig.

DIE LÄNGE DES ARTIKELS

In den letzten Heften waren Artikel zu finden, die drei Ausgaben der SUC Session benötigt haben, um überhaupt komplett erscheinen zu können. Diese Artikel müssen aufgeteilt werden. Mir fällt es ungeheuer schwer, in solch einem Artikel die Abschnitte zu finden, an denen ich sie aufteilen kann. Ist so ein Artikel 20 Seiten lang, muss er zwingen in mindestens drei, besser vier Teile geteilt werden.

Ich habe grundsätzlich keine Probleme mit langen Artikeln, allerdings müssen sie maximal acht Seiten aufgeteilt werden. Der Autor kann das besser als ich. Wenn ich die Teilung vornehmen muss, besteht die Gefahr, dass der Text die Konsistenz verliert und die Lesbarkeit darunter leidet.

FORMULIERUNG UND STIL DER TEXTE

Hier gibt es wenig zu sagen. Im Allgemeinen lasse ich die Texte so, wie sie sind. Es gibt natürlich Ausnahmen, nämlich wenn ich den Text lese und dabei annehmen muss, dass außer mir keiner den Text zu Ende lesen will. In dem Fall weise ich den Text an den Autor zurück und bitte ihn, den Text umzuformulieren.

Schreibt so, wie ihr es auch lesen wollt. Wenn der Text fertig ist, lasst ihn zwei Wochen liegen und lest ihn dann noch einmal durch. Es lohnt sich! Rechtschreibfehler werden erkannt, Formulierungen überarbeitet, Kommata (das ist das Plural von »Komma« - echt wahr!) werden gesetzt, sofern sie nicht vorhanden sind, und so weiter… Versucht es einmal…

KOMMASETZUNG

Kommata, auch *Kommas* genannt, trennen einen Hauptsatz von einem Nebensatz. Sie erleichtern den Lesefluss und vermeiden Irrtümer.

Beispiel:

Komm, wir essen, Opa!

Gegenüber

Komm, wir essen Opa!

Ein kleiner aber feiner Unterschied.

EIN LEIDIGES THEMA: »DASS«« UND »DAS««

»Das« als Artikel, so kennt es jeder:

Das Microdrive, das Interface etc…

Ich habe noch nicht erlebt, dass das irgendjemand falsch geschrieben hat.

»das Thomas« oder *»das Sinclair Spectrum«* wäre nicht korrekt. *»Das«* als Pronomen: *Das Interface, das am ZX Spectrum angeschlossen ist.* Hier gibt es

auch keine Fehler, zumindest habe ich noch keine entdeckt.

Schwierig wird es, wenn aus dem *»das«* ein *»dass«* wird. In diesem Fall ist das *»dass«* eine *Konjunktion*, auch *Binde-* oder *Fügewort* genannt. Mit einer Konjunktion werden Wörter, Satzteile oder Sätze miteinander zu einem Satz verbunden. *»Dass«* leitet dabei meistens einen Nebensatz ein. Es besteht zwischen den einzelnen Satzteilen, also zwischen Haupt- und Nebensatz, eine logische Beziehung. Tatsächlich entstehen hier die meisten Fehler!

Im schwäbischen, also der Gegend, aus der ich komme, kann das *»das«* durch ein schwäbisches *»des«* ersetzt werden – beim Sprechen möglichst lang gedehnt und so undeutlich wie möglich ausgesprochen, eben schwäbisch...

Für alle anderen kann folgender Trick hilfreich sein:

Kann statt *»das«* ein anderes Wort wie **»welches«, »jenes«, »dieses«** oder das *Nomen* selbst eingesetzt werden, wird *»das«* mit nur einem *»s«* geschrieben.

Vielleicht hilft es ja…

UMLAUTE

Wir sind hier in *Good Ol' Germany*. In dieser Sprache sind Umlaute wie ä/ö/ü und ß selbstverständlich. Bitte verwendet sie!

Wenn ich einen Text bekomme, in dem die Umlaute durch ihre Äquivalente *ae/oe/ue* bzw. *ss* ersetzt sind, bedeutet das für mich außerordentlich viel Arbeit, sie zu ersetzten. Außerdem passt ein Doppel-S nicht in jedem Fall, da die lautsprachliche Anwendung im gegebenen Fall eine Andere ist.

BILDER:

Ich kann nur relativ schwierig RAW-Bilder, also Bilder im jeweiligen Roh-Format der entsprechenden Digitalkamera, verarbeiten. Ich muss sie aufwendig umwandeln und stoße bisweilen an meine Grenzen. In der Zwischenzeit habe ich schon einige Tools, die es mir ermöglichen, sie in ein für Word verständliches Format zu konvertieren.

Die Bilder sollten im JPG-Format bei mir ankommen, Komprimierungsfaktor 100%. Zur Information: je kleiner der Prozent-Wert ist, desto stärker ist das Bild komprimiert und umso schlechter ist die Qualität. Die Bildgröße ist weitestgehend egal. Trotzdem bevorzuge die Größe 3888 Pixel in der Breite und 2592 in der Höhe mit 300 dpi Auflösung.

Wenn in den Artikel Bilder hinein kommen sollen, dann beschränkt Euch bitte auf maximal fünf Stück.

REDAKTIONSSCHLUSS

Der Redaktionsschluss ist für mich normalerweise der Zeitpunkt, an dem ich mit der Arbeit am Heft beginne. Je später die Texte bei mir ankommen, desto später kommt das Heft zu Euch und umso größer wird die Hektik für mich. Haltet Euch bitte (!) an den Redaktionsschluss, anderen Falles kommen Eure Beiträge erst in das nächste Heft.

ZUSAMMEN GEFASST

- Dem Artikel eine Überschrift geben.
- Name des Autors/der Autorin angeben, zumindest für den internen Gebrauch.
- Absätze einfügen.
- Bitte keine selbst kreierten Aufzählzeichen, sondern den Tabulator verwenden.
- Lange Texte so teilen, dass sie gegebenen Falles auf mehrere Hefte verteilt werden können.
- Den Artikel so schreiben, wie man ihn selbst gerne lesen möchte.
- Komma einfügen.
- »Das« und »dass« beachten
- Deutsche Umlaute in deutschen Artikel.
- Keine RAW-Bilder sondern JPG.
- Redaktionsschluss beachten.

ZUM SCHLUSS

Leider musste ich schon Texte zurückweisen. Das sind aber äußerst seltene Ausnahmen und keinesfalls die Regel. Entweder hatten sie eine solche Länge, dass sie den Rahmen des Heftes gesprengt haben, oder sie waren in einem Stil geschrieben, der es schwierig machte, sie zu lesen.

Es ist mir immer sehr unangenehm ist, Texte zurück zuweisen, die Alternative ist, dass ich sie umschreibe, wozu ich ehrlicherweise keinerlei Lust habe . In jedem Artikel steckt normalerweise viel Arbeit. Das weiß ich, immerhin schreibe ich selbst genug davon. Tatsächlich schreibe ich selbst Texte, die von anderen Autoren lektoriert werden. Mir gefällt es auch nicht, wenn dieser Lektor, dem ich eine Story übergeben habe, meinen Text umformuliert und Passagen streicht. Ich habe schon erlebt, dass die Story an Lesefluss verloren hat, tatsächlich wird sie aber in den meisten Fällen besser.

Ich selbst streiche in den Texten nichts oder nur ganz wenig. Stilistische Korrekturen nehme ich mit größter Vorsicht vor. Normalerweise lasse ich den Text aber so, wie er ist, da er an Originalität und Authentizität verliert, wenn ich streiche oder umformuliere.

Mir geht es darum, die Layout-Arbeit eines Heftes zügig zu beenden, um es möglichst nah am Redaktionsschluss herauszugeben. Wenn ich viel an einem Artikel korrigieren muss, schiebt sich alles nach hinten, außerdem macht es mir dann keinen Spaß mehr. Immerhin ist es ein Hobby und kein Beruf; aber letztendlich mache ich es gerne. Wenn sich die Autoren an die Vorschläge halten, die ich gemacht habe, bleibt mir auch der Spaß erhalten.

In diesem Sinne!
Euer Redax Joachim

WIE HACKE ICH EIN SPIEL: MONJAS

VON THOMAS EBERLE

Hallo zusammen!

Mein letzter Artikel zu diesem Thema kam wohl gut an, so dass ich diesmal wieder einen aktuellen Hack beschreiben will. Da ich die SCENE+ Diskette erstelle, bleibt es nicht aus, das ich zu diesem Zweck etwas tiefer eintauchen muss in die Materie, dennoch ist dies ein verhältnismäßig einfacher Hack. Ich möchte euch hier zeigen wie einfach es sein kann und dazu animieren, es auch selbst zu versuchen.

Der Hintergrund: Wer ein Divide oder ähnliches benutzt, der kann einfach TAP-Files aus dem Internet herunterladen und abspielen. Wer aber noch einen normalen Spectrum mit Disketteninterface benutzt, muss die Programme umschreiben, damit diese von Diskette laden. Nicht immer ist das einfach, den man muss auch die Einschränkungen der verschiedenen Diskettensysteme beachten.

Am einfachsten und auch immer mein erster Schritt ist die Übertragung auf das *MB02*. Da ich am *MB02* ein SD-Karteninterface habe, kann ich einfach die SD-Karte in den PC einstecken und ein Programm übertragen.

Halt, ganz so einfach auch wieder nicht, man braucht dazu eine Software, die sowohl mit dem MS-DOS-Format des PC´s als auch mit dem BS-DOS Format des *MB02* zurechtkommt. Wie schon an anderer Stelle zuvor beschrieben, nutze ich dazu Total Commander mit den entsprechenden Plugins für *MBH* (*MB02 Festplattenformat*) und (hier nicht notwendig) .TAP. Mit dem TAP-Plugin kann ich mir ein Tap File anschauen, ähnlich wie ein gezipptes File. Wie nicht anders zu erwarten, besteht das Programm aus Basic Loader, einem Ladebild und einem Maschinencode-Teil:

```
MONJAS – Basic 65 Bytes
screen.tap – Ladebild 6912 Bytes
monjas_bin – Maschinencode-Teil
Code Adresse 23990, Länge 41540 Bytes
```

Schauen wir uns zuerst das Basic an:

```
 1 PAPER NOT PI: INK NOT PI:
 BORDER NOT PI: CLEAR VAL "23989":
 LOAD "" SCREEN$:
 PRINT AT VAL "8", NOT PI;:
 LOAD "" CODE:
 PAUSE NOT PI:
 RANDOMIZE USR VAL "23990"
```

Das Basic ist schon recht speichersparend angelegt. Man kann sich denken warum. Der RAMTOP ist mit 23990 relativ hoch. Wenn man eine niedrige Zahl hat, ist es für den RAMTOP hoch - das sagt man so, nicht jeder muss es verstehen.

Dieser Wert mit **CLEAR 23989** legt also fest, ab wann das Basic endet und der Maschinencode beginnen kann. Zwischen Systemvariablen und diesem RAMTOP ist recht wenig Platz, aber auf das Problem kommen wir noch.

NOT PI ist eine Rechenformel und bedeutet: 0 (Null). Somit sind die ersten drei Befehle einfach nur sowohl den Vordergrund (Schrift), Hintergrund und den Border auf Farbe 0 (schwarz) zu setzen. **NOT PI** verbraucht weniger Speicher als eine Zahl zu schreiben. Der RAMTOP Befehl »**CLEAR**« legt dann wie oben beschrieben fest wo der Maschinencode beginnen kann. Auch dieser Wert ist als mathematische Formel, tatsächlich verbraucht also **VAL "23989"** weniger Speicher als **23989** zu schreiben, obwohl es auf den ersten Blick nach mehr Zeichen aussieht. Es wird danach das Bild geladen. Der folgende PRINT–Befehl setzt den PRINT-Zeiger nur ein eine bestimmte Stelle, damit die Meldung beim Laden eines Programms den Ladescreen nicht zerstört. Hierfür gibt es andere Befehle die die Meldung abschalten, aber vermutlich ist auch hier wieder die günstigste Lösung für den Speicherplatz gewählt worden. Dann wird der Maschinencode-Teil geladen.

PAUSE NOT PI bedeutet schlicht ein **PAUSE 0**, also warten auf einen Tastendruck. Danach startet das Maschinencode-Programm.

Das *MB02* Disketteninterface ist das perfekte Interface, den es emuliert quasi die Kassette auf einem anderen Medium wie Diskette oder hier SD Karte. Trotzdem mache ich die *MB02*-Versionen immer so, dass es auch einfach adaptierbar ist für andere Systeme. Klar, sonst hätte ich ja die doppelte Arbeit, den ich setze immer die Programme für *MB02*, *+D*/*Disciple* und die *Opus Discovery* um.

Nicht Pflicht beim *MB02*, aber bei anderen Diskettensystemen ist es, dass ein Ladebefehl immer auch einen Programmnamen haben muss. Da ich schon weiß, dass der Speicher knapp wird, setze ich kurze Programmnamen ein. Das Bild nenne ich **M$**, den Codeteil **Mc**. Speicher will ich aber nicht nur beim Basic sparen, auch werden beide Programmteile gepackt. Für das Bild verwende ich das Utility »SCREEN COMPRESSOR«. Es packt das Bild in weniger als die Hälfte der Bytes, somit verbraucht es nur noch 3238 Bytes. Dafür wird das Bild aber nicht mehr direkt in den Bildschirmspeicher geladen, sondern auf Adresse 50000 und von dort auf-

gerufen. Ich muss also ein **RANDOMIZE USR 50000** einbauen, wieder wird der Basic-Teil länger.

Den Codeteil packe ich mit »TURBO IMPLODER«.

Die Eingaben sind recht einfach, man gibt den Start des MC an, die Länge und die Einsprungadresse, in dem Fall ist Start des MC und Einsprungadresse dieselbe. TURBO IMPLODER packt das Ganze in nur noch 23041 Bytes und es wird kein zusätzlicher Basic Befehl benötigt. Da der Packer immer einige Sekunden braucht zum entpacken, entferne ich den **PAUSE 0**-Befehl. Außerdem lösche ich den **PRINT AT** Befehl, da Diskettenladebefehle keine "BYTES"-Meldung auf den Schirm bringen und so den Screen nicht zerstören (beim *MB02* allerdings schon, da macht es Sinn den Befehl wieder rein zunehmen).

Das BASIC sieht jetzt also wie folgt aus:

```
1 PAPER NOT PI: INK NOT PI:
BORDER NOT PI: CLEAR VAL "23989":
LOAD "M$" CODE:
RANDOMIZE USR VAL "5e4":
LOAD "Mc" CODE:
RANDOMIZE USR VAL "23990"
```

Wie man sieht, habe ich bei dem zusätzlichen **RANDOMIZE**-Befehl auch speicheroptimiert gearbeitet. Der Wert wurde in VAL "" gesetzt und **5e4** bedeutet nichts anderes wie eine 5 mit 4 Nullen, allerdings verbraucht es weniger Zeichen und damit weniger Bytes. Insgesamt ist dieses Basic also sogar noch 5 Bytes kürzer als das Original.

Geschafft... oder?

Natürlich nicht. Den ein RAMTOP auf 23990 ist vielleicht mit dem *MB02* noch machbar, aber *+D/Disciple* und *Opus* benötigen mehr Platz im RAM des Spectrums. Da ist die Fehlermeldung vorprogrammiert. Die Lösung: Wir packen das Ganze auf Adresse 25000. Wie das jetzt? Ganz einfach: Mit **LOAD "Mc" CODE 25000** wird der Code statt auf die ursprüngliche Adresse 23990 eben auf 25000 geladen. Der **CLEAR**-Befehl wird auf 24999 gesetzt... Genug Platz.

Aber läuft ein Programm das auf Adresse 23990 gespeichert war einfach auf einer anderen Adressierung? Theoretisch gibt es Software, die das kann, z.B. wenn nur relative Sprungadressen verwendet werden. Tatsächlich ist aber sicher, dass es nicht gehen wird, da immer feste Sprungadressen auch im Code sind und die dann nicht mehr funktionieren. Zudem ist der Code ja gepackt, im Original war er ja 41540 Bytes lang und hätte damit fast den ganzen Speicher belegt, zu viel um einfach zu verschieben.

Die Lösung: Wir verschieben den Code wieder, nachdem alles im Speicher ist und das Disketten· interface nicht mehr genutzt wird. Dann ist es egal, ob die Systemvariablen des Interfaces überschrieben werden. Dieses einfache Assemblerprogramm bringt den Code wieder an die richtige Stelle:

```
ORG 48042
LD HL, 25000
LD DE, 23990
LD BC, 23041
LDIR
JP 23990
```

Zur Erklärung: Der ORG-Befehl setzt die Adresse, an dem dieses Programm starten soll. In das HL Register laden wir die Adresse, an der die zu kopierenden Daten stehen, in das DE-Register die Adresse an die kopiert werden soll. In BC steht die Länge in Bytes die kopiert werden sollen. LDIR führt dann die Kopieroperation des genannten Blocks aus, genauer gesagt wird der Inhalt von HL auf die Adresse von DE kopiert, dann beide Werte erhöht, der Wert von BC senkt sich um eines und zwar solange, bis BC Null ist. Am Ende folgt der Sprung zum Programmstart.

Dieses Assemblerprogramm gebe ich in meinen Assembler (bei mir PROMETHEUS) ein, assembliere und speichere auf 48042, das ist die Adresse nach dem Programmcode von „Mc“. Beide Codes lade ich zusammen und speichere sie zusammen ab:

```
LOAD "Mc" CODE 25000
LOAD "mon.cd" CODE
```
(das ist das kurze neue Codefile)

```
SAVE "Mc" CODE 25000,23057
```
(Code plus dem kurzen Codefile)

Nun das BASIC noch ändern, die Einsprungadresse ist jetzt nicht mehr 23990, sondern 48042. Ein **CLEAR VAL "23989"**wird nach dem letzten Ladebefehl eingefügt.

Ist es nun geschafft? Leider nicht, ein Programmstart zeigt, dass der **CLEAR VAL "23989"** nicht durchgeführt wird. Das Programm ist durch diesen CLEAR-Befehl zu lang geworden, es passt nicht mehr *unter* den RAMTOP. Aber kürzen kann ich eigentlich nichts mehr Ich könnte die Programmnamen auf einen Buchstaben kürzen, aber auf einer Diskette mit ca. 20 Programmen möchte ich das erst gar nicht anfangen. Das Alphabet hat nur 24 Buchstaben, da gehen die mir bald aus. Ich habe dann das BASIC aufgeteilt. Nach dem **RANDMIZE USR VAL "5e4"** habe ich einen weiteren LOAD Befehl hinzugefügt : **LOAD "Mon.b"**

Den Lade- und Startbefehl des MC habe ich in ein weiteres BASIC – Programm gepackt und mit **SAVE "Mon.b" LINE 1** abgespeichert.

Et voilà, ich habe nun folgende Files:

- *MONJAS* – Basic 51 Bytes
- *M$* - Ladebild als Codefile auf Adresse 50.000 mit 3238 Bytes
- *Mon.b* - Zweites Basic File mit 32 Bytes
- *Mc* – Codefile auf Adresse 25000 mit 23057 Bytes

Insgesamt sind das 26378 Bytes aus ehemals 48517 Bytes, fast die Hälfte.

Das Basic ist insgesamt etwas länger geworden, aber dafür in zwei Files aufgeteilt:

MONJAS:

```
1 PAPER NOT PI: INK NOT PI:
BORDER NOT PI: CLEAR VAL „2499“:
LOAD "M$" CODE:
RANDOMIZE USR VAL „5e4“:
LOAD "Mon.b"
```

Mon.b:

```
1 LOAD "Mc" CODE:
CLEAR VAL "23989":
RANDOMIZE USR VAL "48042"
```

That´s it. Wieder ein Programm mehr auf der Diskette, das Ganze könnt ihr dann auf der **SCENE+** Diskette bestaunen. Übrigens ein sehr lustiges Spiel.

SYSTEMVARIABLEN DES ZXSPECTRUM 48

ARTIKEL AUS YOUR SPECTRUM, ISSUE 2&3, MÄRZ & APRIL 1984

Systemvariablen sind Bytes im Speicher, die dem Spectrum helfen, sich an bestimmte Dinge zu erinnern, die er über sich selbst wissen muss - wenn Sie so wollen, die Haushaltsroutinen. Tauchen Sie mit **DILWYN JONES** tiefer in das Spectrum-ROM ein, in diesem ersten von zwei Artikeln, die den gesamten verfügbaren Satz von Systemvariablen untersuchen und umfassende Richtlinien darüber geben, was Sie mit ihnen tun können und was nicht.

Die Systemvariablen sind Bytes im Spectrum-ROM, die es dem Computer ermöglichen, alle Dinge zu wissen, die er tun sollte, um so zu funktionieren, wie Sie es erwarten. Die Informationen, z. B. wie der Speicher des Spectrum aufgebaut ist, werden in den Systemvariablen an diesen Adressen gespeichert, so dass der Computer sie abrufen und bei Bedarf aktualisieren kann.

Wir können die Informationen, die in diesen Speicherplätzen gespeichert sind, in unseren Programmen nutzen, indem wir entweder bereits vorhandene Informationen lesen oder sie ändern, damit der Computer etwas tut, was er sonst vielleicht nicht tun würde, oder manchmal auch, um es einfacher zu machen.

Nicht alle davon sind für uns von großem Nutzen. Und sicherlich sollten auch nicht alle geändert werden. Manche führen dazu, dass der Computer abstürzt, oder der Computer ignoriert Sie einfach. Einige lassen sich nur unter bestimmten Umständen und in den meisten Fällen nur mit strengen Einschränkungen ändern. Ich hoffe, dass ich Ihnen einige Richtlinien geben kann, was getan werden kann und was nicht, aber hoffentlich werden Sie mit der Zeit auch Ihre eigenen kleinen PEEKs und POKEs kennen lernen.

23552 BIS 23559 KSTATE
LESEN DER TASTATUR

Mit jedem Interrupt des Prozessors, normalerweise 50 Mal pro Sekunde, wird unter anderem die Tastatur ausgelesen und die Ergebnisse hier gespeichert. Die Bytes haben unterschiedliche Verwendungen. Nicht alle können vom Programmierer praktisch verwendet werden. Sie können dieses Programm verwenden, um zu untersuchen, was in den acht Bytes von **KSTATE** vor sich geht. Führen Sie es aus und drücken Sie verschiedene Tasten, um zu sehen, welche Auswirkungen einzelne Tasten haben, z. B. die Umschalttasten, und welche Auswirkungen der Wechsel von einer Taste zur anderen hat.

```
10 FOR A=23552 TO 23559
20 POKE 23692,0: REM KEEP SCROLLING
30 LET B=PEEK A
40 PRINT A; TAB 10;B; TAB 20;
   CHR$ B AND B>31
50 NEXT A
60 GO TO 10
```

Die ersten vier Bytes von **KSTATE** befassen sich mit dem sogenannten *»two key rollover«,* der es Ihnen ermöglicht, eine zweite Taste zu drücken, bevor Sie die erste loslassen. Die Beschreibungen der vier wichtigsten Bytes, 23556 bis 23559, gelten auch für die ersten vier Bytes, sofern Sie bedenken, dass diese nur bei einem *Rollover* mit zwei Tasten zum Einsatz kommen. **PEEK 23556** kann den Code der Großbuchstabenversion der gedrückten Taste zurückgeben, d.h. wenn Sie das *Symbol Shift A* drücken, erhalten Sie den Code von *'A'*, nicht den Code von 'a' oder den Code von *'STOP'*.

Dies kann nützlich sein, wenn Großbuchstaben eingegeben werden müssen, usw. Die Wirkung des Tastendrucks ist vorübergehend und hält nur so lange an, wie die Taste gedrückt wird. Der Wert in 23556 wäre 255, wenn zu dem Zeitpunkt, zu dem der Interrupt aufgetreten ist, keine Taste gedrückt wurde. Für die Eingabetaste wird ein Wert von 13 zurückgegeben. Für die Leertaste wird ein Wert von 32 zurückgegeben.

Das gleichzeitige Drücken beider Umschalttasten ergibt 14. Das folgende Programm soll dies demonstrieren:

```
10 LET A=PEEK 23556
20 POKE 23692,0
30 PRINT A, CHR$ A AND A>31
40 GO TO 10
```

23557 wird für die Zeitmessung verwendet, um zu verhindern, dass ein unregelmäßiger Tastenkontakt usw. zu Problemen führt - auch bekannt als *»Keyboard debouncing«.*

23558 ist der Timer für die automatische Tastenwiederholung, der die Pause vor dem Beginn der Tastenwiederholung und die Pause zwischen den Wiederholungen nach dem Beginn der Tastenwiederholung zählt. Die verwendeten Verzöge-

rungen sind die in den Systemvariablen, die diese Verzögerungen enthalten (23561/2).

23559 enthält den Code des zuletzt auf der Tastatur gedrückten Zeichens. Dies hängt davon ab, ob die Shift-Tasten gedrückt wurden oder nicht. Die erzeugten Zahlen sind die, die von **PRINT CODE INKEY$** zurückgegeben würden, mit dem Unterschied, dass es sich um die zuletzt gedrückte Taste handelt und nicht notwendigerweise um die Taste, die gerade gedrückt wird. Probieren Sie dieses Programm aus, um zu sehen, was passieren kann - führen Sie es aus und versuchen Sie, verschiedene Tasten zu drücken, wobei Sie die Umschalttasten verwenden.

```
10 LET A=PEEK 23559
20 POKE 23692,0
30 PRINT A, CHR$ A AND A>31
40 GO TO 10
```

Siehe auch unter 23611 FLAGS.

23560 LAST_K
NEU GEDRÜCKTE TASTE

Jedes Mal, wenn die Tastatur abgefragt und festgestellt wird, dass eine Taste gedrückt wurde, die sich als gültig erweist, wird der Wert dieser Systemvariablen aktualisiert. Ihr Inhalt ist der Code der zuletzt gedrückten Taste.

Diese Systemvariable tut nicht viel, was man nicht auch mit **INKEY$** tun könnte, außer dass sie dazu benutzt werden könnte, ein Zeichen vorwärts zu tippen. Wenn Sie dieses Programm ausprobieren, werden Sie feststellen, dass, wenn Sie eine Taste drücken, wenn Sie dazu aufgefordert werden, die Taste in kurzer Zeit auf dem Bildschirm angezeigt wird, auch wenn das Programm vielleicht noch nicht bis zur Zeile 50 gekommen ist, als Sie eine Taste gedrückt haben. Der Code der zuletzt gedrückten Taste wird hier gespeichert und bleibt hier, bis eine weitere Taste gedrückt wird. Es ist möglich, zu testen, ob eine Taste gedrückt wurde, indem man Bit 5 der Systemvariablen *FLAGS 23611* untersucht. Dieses Bit wäre '1' für eine gerade gedrückte Taste.

```
10 PRINT "Drücke jetzt eine Taste"
20 FOR A=1 TO 900
30 NEXT A
40 CLS
50 LET A=PEEK 23560
60 PRINT A: IF A>31 THEN PRINT CHR$ A
```

Dies könnte zum Testen einer y/n (ja oder nein) Situation verwendet werden - wenn Sie wissen, dass eine solche Situation bevorsteht, könnten Sie Ihre Antwort angeben, bevor das Programm dazu kommt, und das Programm würde antworten, wenn es dazu kommt. Auch wenn zwei Tasten gleichzeitig gedrückt werden, reagiert das Programm, wenn eine davon losgelassen wird, ohne dass man warten muss, bis die Tastatur vollständig losgelassen wird.

Steuerzeichen können mit Caps Shift in Verbindung mit den Zifferntasten erzeugt werden. Die Eingabetaste gibt 13 zurück. Wenn Sie beide Umschalttasten gleichzeitig drücken, wird 14 ausgegeben. Um dies zu sehen, probieren Sie dieses Programm aus:

```
10 LET A=PEEK 23560
20 PRINT A, CHR$ A UND A>31
30 GO TO 10
```

23561 REPDEL
WIEDERHOLUNGSVERZÖGERUNG

Diese Systemvariable enthält die Zeitspanne, die eine Taste gedrückt gehalten werden muss, bevor die automatische Wiederholung beginnt. Die Zeitverzögerung beträgt eine fünfzigstel Sekunde und beginnt bei 35/50 Sekunden. Sie können diese Zeitspanne gerne ändern, z.B. **POKE 23561,1** wenn Sie z. B. möchten, dass die Taste sofort mit der Wiederholung beginnt. Die Cursor sind allerdings etwas schwieriger zu steuern. **POKE 23561,0** schaltet die automatische Wiederholung effektiv aus und führt zu einer Verzögerung von etwa fünf Sekunden, ebenso wie **POKE 23561,255.**

23562 REPPER
VERZÖGERUNG ZWISCHEN WIEDERHOLUNGEN

Diese Systemvariable steuert die Zeitspanne zwischen den Wiederholungen, sobald die automatische Wiederholung tatsächlich begonnen hat. Die Zeit wird in fünfzigstel Sekunden angegeben. Wenn Sie die automatische Wiederholung aus irgendeinem Grund abschalten wollen, können Sie mit **POKE 23562,0** oder POKE **23562,255** einen Abstand von etwa fünf Sekunden zwischen den Wiederholungen einstellen. Wenn Sie lange Programmzeilen bearbeiten wollen (z.B. eine lange PRINT-Anweisung), dann beschleunigt **POKE 23562,1** die Bewegung des Cursors an die richtige Stelle. Achten Sie aber darauf, 23562 nicht zu oft zu ändern, sonst wird der Cursor so schnell, dass er nur noch schwer zu kontrollieren ist. Der normale Wert ist 5/50 bis eine Sekunde oder eine Zehntelsekunde.

23563/4 DEFADD

Die Adresse des Arguments einer benutzerdefinierten Funktion in einem Programm; d.h. wenn Sie **DEF FN A(B)** in einer Programmzeile haben, wäre der Wert in 23563/4 die Adresse des Buchstabens B in den Klammern in dieser Zeile, während nur die Funktion verwendet wird.

Der beste Weg, dies mit **PEEK in 23563/4** zu zeigen, ist, das **PEEK** als Teil der auszuwertenden FN zu setzen, da dort immer eine Null steht, es sei denn, die Funktion wird ausgewertet. Also die Zeile:

```
10 PRINT PEEK 23563+256*PEEK 23564
```

würde immer Null zurückgeben. Auf der anderen Seite:

```
10 DEF FN A(B)=PEEK 23563 +
   256*PEEK 23564
20 PRINT FN A(999)
```

würde die Adresse von B in Zeile 10 zurückgeben. Die 999 ist nicht von Bedeutung, sondern

dient nur dazu, B einen Wert zuzuweisen, um einen Fehler zu vermeiden. Im Falle einer Funktion ohne Argument:

```
10 DEF FN A()=SPITZE 23563 +
   256*PEEK 23564
20 PRINT FN A()
```

Dies würde die Adresse des Symbols der geschlossenen Klammer ausgeben.

23568 BIS 23605 STRMS

Die ersten 14 Bytes enthalten Adressen, die sich auf Kanäle und Streams beziehen. Die Streams -3 bis +3 werden in jeweils zwei Bytes gespeichert.

23606/7 CHARS

Diese Systemvariable hat als normale Werte:

23606 enthält **0**
23607 enthält **60**

Diese Systemvariable zeigt auf den Anfang des Zeichensatzes, den der Computer für die Ausgabe auf dem Bildschirm und dem Drucker verwendet. **SCREEN$** verwendet ebenfalls diese Systemvariable. Die normale Adresse, auf die sie zeigt, ist 15360, was 256 weniger ist als die Adresse des Beginns des ROM-Zeichensatzes. 256 weniger, weil auf den Zeichengenerator durch etwas Ähnliches wie

```
PEEK 23606+256*PEEK 23607+CODE "A"*8
```

zugegriffen wird, und da das erste Zeichen ein Leerzeichen ist, und der Code eines Leerzeichens 32 ist, kann man sehen, dass 8 * 32 256 ist.

Der Zeichengenerator ist 768 Bytes lang. Wenn Sie also einen neuen Zeichensatz einrichten wollen, müssen Sie diese Anzahl von Bytes sichern, für den Fall, dass er von Basic überschrieben wird. Wird der Wert überschrieben, wäre der Zeichensatz nur Kauderwelsch, der Computer würde aber weiter laufen und nicht abstürzen.

Es wurde erwähnt, dass **SCREEN$** diese Systemvariable verwendet - in der Tat ist Ihnen vielleicht das Problem bekannt, dass **SCREEN$** normalerweise keine benutzerdefinierten Grafiken erkennt, es sei denn, sie ähneln ein bestehendes Spectrum-Zeichen. Tatsächlich arbeitet **SCREEN$**, indem es die Adresse des Anfangs des Zeichengenerators aufnimmt und die Tabelle durchsucht, bis es ein passendes Zeichen findet. Da der Spectrum-Bildschirm nicht wie bei anderen Computern im Speicher, sondern in *Bits* abgebildet wird, bleibt ein einmal auf den Bildschirm gedrucktes Zeichen auch dann erhalten, wenn man es im Speicher ändert. Wir könnten also den Zeiger auf den Zeichensatz vorübergehend so ändern, dass er auf die benutzerdefinierte Grafik zeigt, und dort nachsehen.

Ein Problem besteht darin, dass es zwar eine Systemvariable gibt, die uns sagt, wo die benutzerdefinierten Grafiken beginnen, diese Adresse aber 256 oder mehr beträgt - wir müssen also 256 subtrahieren. Das bedeutet, dass wir vom High-Byte eine Eins abziehen. Dieses Programm sollte dies demonstrieren:

```
10 FOR X=144 TO 164
20 PRINT AT 0,0; CHR$ X
30 POKE 23606, PEEK 23675
40 POKE 23607, PEEK 23676-1
50 PRINT AT 20,0; SCREEN$ (0,0)
60 PAUSE 40
70 POKE 23606,0
80 POKE 23607,60
90 NEXT X
```

Wir haben dafür gesorgt, dass der Computer die benutzerdefinierten Grafiken für den normalen Zeichensatz hält. **SCREEN$** wird immer noch Zeichen mit den Codes 32-127 erzeugen, obwohl dies mit ein wenig Tüftelei leicht zu umgehen ist. Da **SCREEN$** mit **CHR$ 32** beginnt und die UDGs mit 144 beginnen, müssten wir 112 hinzufügen, um Zeichen im Bereich der benutzerdefinierten Grafiken auszugeben.

Hier ist eine Möglichkeit, dies zu tun. X ist die x-Koordinate auf dem Bildschirm und Y ist die y-Koordinate auf dem Bildschirm an der Stelle, die **SCREEN$** untersuchen soll. Es wird zunächst geprüft, ob **SCREEN$** dort nicht eines der normalen Zeichen findet, und dann zurückgegeben, wenn eines gefunden wird. Das Zeichen an Y,X wird in A$ zurückgegeben. Zeile 8025 wird nur benötigt, wenn Sie einen anderen Zeichensatz als den ROM-Zeichensatz verwenden. Wenn Sie den ROM-Zeichensatz verwenden, löschen Sie Zeile 8025 und ersetzen Sie die Zeilen 8070 und 8080 durch die folgenden alternativen Versionen.

```
8000 REM SCREEN$ FOR UDGs
8010 LET A$=SCREEN$ (Y,X)
8020 IF A$()="" THEN RETURN
8025 LET A=PEEK 23606: LET B=PEEK 23607
8030 POKE 23606, PEEK 23675
8040 POKE 23607, PEEK 23676-1
8050 LET A$=SCREEN$ (Y,X)
8060 IF A$()<>"" THEN LET
     A$=CHR$ (CODE A$+112)
8070 POKE 23606,A
8080 POKE 23607,B
8090 RETURN

8070 POKE 23606,0
8080 POKE 23607,60
```

Die Geschichte ist damit noch nicht zu Ende. Es gibt nur 21 benutzerdefinierte Grafiken - wenn **SCREEN$** keine Übereinstimmung findet, sucht es weiter über die benutzerdefinierten Grafiken hinaus, bis es den Bereich von 32 bis 127 gefunden hat, nach dem es zu suchen glaubt. Dies könnte peinlich sein, wenn aus irgendeinem Grund Daten oberhalb der UDGs gespeichert sind, die einem beliebigen Zeichen ähneln. Um dies zu verhindern, obwohl die UDGs normalerweise sowieso ganz oben im RAM liegen, könnte diese Zeile hinzugefügt werden:

```
8065 IF A$>CHR$ 164 THEN LET A$=""
```

Übrigens sollten Sie sicherstellen, dass 23606/7 immer auf den richtigen Zeichensatz zeigt, wenn **PRINT**, **LIST** usw. durchgeführt wird.

23608 RASP
DAUER DES FEHLERMELDETONS

Steuert die Dauer des Signaltons, der ertönt, um Sie zu warnen, dass der Speicher knapp wird. Beim Einschalten hat dies einen Wert von 64. Dieser Wert kann geändert werden, aber es scheint wenig Sinn zu machen. **POKE 23608,0** ergibt ein sehr kurzes Klicken anstelle eines Signaltons - nützlich, wenn Sie den Signalton hassen, der Sie satirisch verhöhnt, wenn Ihnen der Speicher ausgeht. Alternativ gibt **POKE 23608,255** einen sehr langen Ton aus, der die Tastatur lahmlegt und Sie daran hindert, weiter als bis zum Ertönen des Tons zu tippen.

23609 PIP
KLICKLÄNGE BEI TASTENDRUCK

Steuert die Länge des Klicks, der beim Drücken einer Taste im Befehlsmodus oder während eines **INPUT** ertönt. Der Wert beginnt bei Null, kann aber geändert werden. Jeder Wert zwischen 30 und ca. 130 ergibt einen angenehmen, hörbaren Piepton anstelle des üblichen leisen Klicks. Bei Werten über 130 verlangsamt sich die Reaktion der Tastatur merklich (da die Berechnung beim Ertönen des Pieptons stoppt). Normalerweise wird **POKE 23609,100** verwendet.

23610 ERR_NR
FEHLERCODE-1

Steuert die Nummer des Fehlerberichts und hat normalerweise einen Wert von 255, es sei denn, es tritt ein Fehler auf. Dann enthält er einen Wert weniger als die gedruckten Fehlercodes, z. B. für Fehler 4, Speicherplatzmangel, würde er eine Drei enthalten. Die ausgedruckte Meldung befindet sich im ROM ab der Adresse 5010 dezimal. Das Ende der Meldung wird dadurch angezeigt, dass Bit 7 des letzten Zeichens der Meldung auf eine Eins gesetzt ist. Nach der Fehlermeldung kommt die '© 1982 Sinclair Research Ltd.'-Meldung, die Sie nach dem Einschalten oder NEW sehen. Sie können mit POKE 23610 eine Fehlermeldung erzeugen, um das Programm zu stoppen, aber da die gedruckte Meldung feststeht und im ROM gespeichert ist, kann es sein, dass Sie am Ende Müll erhalten.

Wenn Sie einen **'out of memory'**-Fehler simulieren wollten, würden Sie ein Programm mit **POKE 23610,3** beenden. Dies würde nicht als beliebige Programmzeile funktionieren; man müsste sicherstellen, dass es die letzte Zeile ist, denn sobald eine Bedingung zum Beenden eines Programms eintritt, wird 23610 erst dann betrachtet, um zu bestimmen, was ausgegeben wird.

23611 FLAGS

Diese Systemvariable enthält verschiedene Flags zur Steuerung des Basic-Systems und sollte im Allgemeinen nicht ge**POKE**ed werden. Einige der Flags können jedoch sinnvollerweise ge**PEEK**ed werden.

Bit 0: Eine Eins bedeutet, dass vor dem nächsten Schlüsselwort kein Leerzeichen gedruckt werden soll.

Bit 1: Wenn dieses Bit auf eine Eins gesetzt ist, bedeutet dies, dass die Druckausgabe an den Drucker geschickt werden soll. Eine Null würde bedeuten, dass sie an den TV-Bildschirm gesendet werden soll.

Bit 5: Jede neu gedrückte Taste wird dadurch angezeigt, dass ihr Code in 23560 (der Systemvariablen LASTK) gespeichert wird und Bit 5 von 23611 (FLAGS) gesetzt wird, um anzuzeigen, dass eine neue Taste gedrückt wurde.

Bit 7: Syntax-Flag.

Diese sind von größerem Nutzen, wenn ROM-Routinen in einem Maschinencode-Programm verwendet werden.

23613/4 ERR_SP

Verfolgt die Adresse auf dem Maschinenstapel, dem *Stack*, wo die entsprechenden Rückgabedaten liegen. Versuchen Sie, ein paar **GOSUBs** ohne passende **RETURNs** aufzurufen, und beobachten Sie diesen Punkt im Speicher. Jetzt können Sie sehen, was passiert und warum dies geschieht, wenn Ihnen in einer solchen Situation der Speicher ausgeht. Versuchen Sie auch, den Inhalt der drei Adressen (auf deren Basis 23613/4 zeigt) mit **PEEK** abzufragen, um zu sehen, woraus die Rückgabedaten tatsächlich bestehen.

```
10 LET A=PEEK 23613 + 255*PEEK 23614
20 PRINT PEEK A; TAB 10; PEEK (A+1);
   TAB 20; PEEK (A+2)
```

23617 MODUS
CURSOR-MODE

Gibt den Cursor an. Die Werte Null, Eins, Zwei oder Vier geben den L/C-Modus, E-Modus, G-Modus bzw. K-Modus an. Das Anklicken dieser Systemvariablen beeinflusst das Aussehen des Cursors - er kann als blinkender Buchstabe, Zahl, Symbol oder sogar als Keyword erscheinen. Am deutlichsten wird dies während einer **INPUT**-Anweisung. Der Wert wird bei Bedarf zurückgesetzt, z. B. bei einem Moduswechsel, der normalerweise über die Tastatur erfolgt. Wenn Sie also Schwierigkeiten haben, drücken Sie beide Umschalttasten, um in den E-Modus zu wechseln, und dann noch einmal die gleiche Taste, um in den normalen L/C-Modus zurückzukehren.

Probieren Sie dieses Programm aus, das alle möglichen Werte in 23617 eintippt. Die meisten sind Varianten für die vier Cursors, d.h. Sie werden sich nach dem **POKE** in einem bestimmten Modus wiederfinden, z.B. alles kommt als Grafik wie im G-Modus. 252 ergibt einen blinkenden L/C-Modus '<', der auf die Stelle zeigt, an der man gerade tippt ...

```
10 FOR A=0 TO 255
20 PRINT A
```

```
30 POKE 23617,A
40 INPUT A$
50 NEXT A
```

Indem Sie verschiedene Werte in diese Systemvariable POKE eingeben, können Sie einen blinkenden, hellen, mehrfarbigen unteren Bildschirm erzeugen oder sowohl PAPER als auch INK in derselben Farbe darstellen, um zu verhindern, dass andere Personen an Ihre Programme herankommen - jede Änderung müsste jedoch blind erfolgen. Sie könnten auch die INPUTs besonders hell machen, um sich abzuheben.

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Lower screen FLASH	lower screen BRIGHT	BORDER colour und lower screen PAPER			lower screen INK		

Tabelle 1. Die Bits der Systemvariablen 23624 steuern die Eigenschaften des unteren Bildschirms und die Farbe BORDER

23618/9 NEWPPC UND 23620 NSPPC
ZEILENNUMMER FÜR EINSPRUNG

23618/9 ist eine Zwei-Byte-Systemvariable, die die Zeilennummer der Zeile enthält, zu der gesprungen werden soll. 23618 enthält das niedrigere Byte der Zeilennummer und 23619 das höhere, so dass die enthaltene Zeilennummer als **PEEK 23618 + 256 * PEEK 23619** gelesen wird. Um eine Zeilennummer einzugeben, sagen wir Zeile X:

```
POKE 23618, X-256*INT (X/256)
POKE 23619, INT (X/256)
```

Wir kommen nun zur Systemvariablen 23620. Mit 23618/9 und 23620 könnten wir tatsächlich einen **GO TO**-Befehl an eine Anweisung innerhalb einer Programmzeile simulieren, falls das einmal nötig sein sollte. **GO TO**s können nicht auf einzelne Anweisungen innerhalb langer Programmzeilen zugreifen.

Um zur Anweisung 4 in Zeile X zu springen, gehen Sie zunächst die oben beschriebenen Schritte durch, dann **POKE 23620,4** und der Sprung wird ausgeführt.

23624 BORDCR
BORDER-COLOUR

Die Bits dieser Systemvariablen steuern die Eigenschaften des unteren Bildschirms und die Farbe **BORDER**. Werfen Sie einen Blick auf *Tabelle 1* auf dieser Seite.

Indem Sie verschiedene Werte in diese Systemvariable poken, können Sie einen blinkenden, hellen, mehrfarbigen unteren Bildschirm erzeugen oder sowohl **PAPER** als auch **INK** in der gleichen Farbe darstellen, um zu verhindern, dass andere Personen an Ihre Programme herankommen - allerdings müsste jede Änderung blind erfolgen. Sie könnten auch die **INPUT**s besonders hell machen, um sie hervorzuheben.

23627/8 VARS
ZEIGER BEGINN VARIABLENSPEICHER

Der Zeiger auf den Anfang des Variablenspeichers. Abgesehen davon, dass Sie den Weg in den Variablenbereich finden, können Sie mit diesem Ausdruck die Länge des Basic-Programms ermitteln. Bildschirm, Systemvariablen, Stacks und Variablen werden dabei nicht berücksichtigt.

```
LET bytes=PEEK 23567+256*PEEK 23628-
PEEK 23635-256*PEEK 23636
```

23629/30 DEST
VARIABLENADRESSE BEI ZUWEISUNG

Die Adresse der Variablen, wenn sie zugewiesen wird. Wenn die Variable bereits vorher eingerichtet wurde, zeigt sie auf den Anfang des Bereichs, in dem sie im Variablenbereich gespeichert war. Wird sie zum ersten Mal definiert, zeigt sie auf die Adresse des Anfangs des Variablennamens im Programm, z. B. in

```
10 LET A = 5
```

auf die Adresse des Buchstaben A.

Es kann auch verwendet werden, um die Speicheradresse einer numerischen Variablen zu finden, wenn Sie etwas wie **LET A=A** wie im folgenden Programm verwenden:

```
10 LET A=5
20 LET A=A
30 PRINT PEEK 23629+256*PEEK 23630
```

23631/2 CHANS
ADRESSE DER ERSTEN KANALINFO

Speichert die Adresse, an der der Kanalinformationsbereich beginnt.

23633/4 CURCHL
ADRESSE DES AKTUELLEN INPUT/OUTPUT

Die Adresse der zu diesem Zeitpunkt verwendeten **INPUT/OUTPUT**-Informationen. Normalerweise zeigt sie während einer **INPUT/OUTPUT**-Operation auf einen 5-Byte-Datenblock im Kanalinformationsbereich. Verwenden Sie dieses Programm, um den Inhalt zu prüfen.

```
1 FOR X=0 TO 3:
  PRINT #X; PEEK 23633+256*
  PEEK 23634: NEXT X: PAUSE 0: STOP
```

Hier beginnt der zweite Teil des Artikels aus

YOUR SPECTRUM, May 1984

23635/6 PROG
BEGINN DES BASIC-SPEICHERS

Die Adresse des Anfangs des Bereichs im Speicher, in dem das Basic-Programm gespeichert ist. Diese Adresse zeigt auf das erste Byte der Zeilennummer der ersten Programmzeile. Dies kann

nützlich sein, wenn Sie Programme für andere Sinclair-Research-Computer konvertieren, deren Informationen als **REM**-Anweisung in der ersten Zeile eines Programms stehen. Siehe auch unter VARS oben.

Wenn Sie eine Zeile in einem Programm "sichern" wollen, können Sie mit Hilfe dieser Systemvariablen eine Null in beide Bytes einer Zeilennummer am Anfang eines Programms einfügen. Programmzeilen beginnen mit einer zwei Byte langen Zeilennummer.

23637/8 NXTLIN
BEGINN NÄCHSTE PROGRAMMZEILE

Die Adresse des Beginns der nächsten Programmzeile. Damit können Sie auf Maschinencode zugreifen, der in REM-Anweisungen an beliebiger Stelle im Programm gespeichert ist, z. B. auf solche, die mit MERGE aus einer Bandbibliothek von Unterprogrammen geladen wurden. Diese würden ihre eigenen lokalen Aufrufe an den Maschinencode haben, etwa so:

```
9000 LET A=USR (PEEK 23637+256*
     PEEK 23638+5)
9010 REM <> MACHINE CODE <>
9020 RETURN
```

Eine Einschränkung besteht darin, dass in die **REM**-Anweisung keine Steuerzeichen für Farbe, Blinken, Helligkeit usw. eingefügt werden dürfen, da sie sonst als Maschinencode interpretiert werden könnten, was die Sache etwas durcheinander bringt. Bei der Verwendung von Unterprogrammen aus einer Bibliothek würden diese jedoch normalerweise ohnehin nicht verwendet werden.

23639/40 DATADD
ADRESSE LETZTES KOMMA VON 'DATA'

Hier steht die Adresse des Kommas, das die letzten Daten beendet. Wenn nichts aus der Liste gelesen wurde (z.B. nach **RUN** oder **RESTORE** usw.), ist die Adresse in 23639/40 die Adresse des Bytes vor dem Programmbereich, normalerweise die **CHR$ 128** am Ende des Kanalinformationsbereichs. Zur Veranschaulichung versuchen Sie, dieses Programm zu starten:

```
10 DATA "1","2","3","4","5"
20 LET A=PEEK 23639 + 256*PEEK 23640
30 PRINT A; TAB 9; PEEK A; TAB 18;
   CHR$ PEEK A AND PEEK A>31
40 READ B$
50 GO TO 20

23754 128
23763 44,
23767 44,
23771 44,
23775 44,
23779 13
```

Die Adresse in dieser Zwei-Byte-Systemvariablen kann auf das Enter-Zeichen oder den Doppelpunkt verweisen, der das Ende der Zeile oder Anweisung, die die Daten enthält, kennzeichnet - die Adresse des Terminators des letzten Datenelements.

23641/2 E_LINE
BEGINN DES BEREICHS ÜBER DEN VARIABLEN

Diese Systemvariable verweist auf den Beginn des Bereichs über den Variablen. Daraus lässt sich ablesen, wie viel Speicher in Bytes vom Bildschirm, den Systemvariablen, dem Programm und den Variablen verbraucht wird, nachdem das Programm zum Einrichten der Variablen usw. ausgeführt wurde. Geben Sie dies als direkten Befehl ein:

```
PRINT PEEK 23641+256*
PEEK 23642-16384
```

Wir können auch feststellen, wie viel Platz für Variablen verwendet wird, nachdem das Programm zum Einrichten der Variablen mit **RUN** gestartet wurde. Verwenden Sie den Befehl:

```
PRINT PEEK 23641+256*PEEK 23642-
PEEK 23627 - 256*PEEK 23628
```

23658 FLAGS2

Diese Systemvariable enthält einige Flags, die (normalerweise) vom Computer verwendet werden, um bestimmte Bedingungen anzuzeigen.

Am besten können wir sie nutzen, um das durch Bit 3 angezeigte Flag zu verwenden. Eine Eins zeigt an, dass die Feststelltaste eingeschaltet ist oder die Feststelltaste aktiviert ist.

Wozu ist das gut? In einem Programm, das **INKEY$** verwendet, z. B. in einem Menü mit Optionen in einem Ablageprogramm müssen wir oft wissen, ob der Bediener eine bestimmte Taste drückt. Wenn der Bediener aufgefordert wird, »**Y**« für **Ja** oder »**N**« für **Nein** zu drücken, kann es sein, dass er »**y**« für **Ja** oder »**n**« für **Nein** drückt - und dabei Klein- und Großbuchstaben verwechselt. Meistens hängt dies davon ab, ob die Feststelltaste aktiviert ist - Menschen interessieren sich nicht für Groß- oder Kleinschreibung, und ob sie »**y**« oder »**Y**« drücken, erwarten sie, dass der Computer dies wie ein Mensch versteht. Aber der Computer weiß das nicht wirklich zu schätzen. Wenn wir also automatisch die Feststelltaste aktivieren, sind unsere Sorgen vorbei und wir haben ein einfacheres Programm, das nicht für jede Auswahl zwei verschiedene Optionen prüfen muss.

Es ist verlockend, die Basic-Anweisung **POKE 23658,8** zu verwenden, um die Feststelltaste zu aktivieren und **POKE 23658,0**, um sie zu deaktivieren. Dies hat jedoch Auswirkungen auf die anderen Flags, so dass Sie deren Status zuerst überprüfen sollten, es sei denn, Sie wissen, dass sie keinen bestimmten Wert haben. Normalerweise hat 23658 im **L**-Modus den Wert Null, so dass es im Allgemeinen OK ist, die obigen **POKE**s zu verwenden. Sie werden wahrscheinlich keine Abstürze verursachen, aber in seltenen Fällen kann es zu komischen Effekten kommen. Wenn der Druckerpuffer leer ist, wird Bit 1 auf Null gesetzt.

23659 DF_SZ
ANZAHL DER ZEILEN UNTERER BILDSCHIRM BEREICH

Diese Systemvariable enthält die Anzahl der Zeilen im unteren Teil des Bildschirms, der normalerweise für I**NPUT**-Meldungen, Fehlerberichte usw. verwendet wird. Normalerweise ist dies eine Zwei, außer wenn ein langer **INPUT**-Prompt angezeigt wird, usw. Wenn ein Wert von Null eingegeben wird, normalerweise um zu versuchen, diesen ungenutzten Teil zu löschen, damit wir die gesamten 24 Zeilen des Bildschirms nutzen können, stürzt der Computer ab.

Dies ist jedoch nur mit einigen Einschränkungen möglich. So muss sichergestellt werden, dass der untere Teil des Bildschirms wieder in den Normalzustand versetzt wird, bevor dieser genutzt werden kann - ein Programmabbruch wäre also ziemlich katastrophal! Auch Fehler, die im Laufe eines Programms auftreten, haben den gleichen Effekt, da der Fehlerbericht ausgedruckt werden muss.

Hier ist ein kurzes Listing, um die Verwendung der Zeilen 22 und 23 auf dem Bildschirm zu demonstrieren. Leider funktioniert es nur für PRINT oder **PRINT TAB**, da wir **PLOT** hier nicht verwenden können und PRINT AT nur bis zur Zeile 22 funktioniert. Der Bildschirm wird durch **POKE 23659,2** innerhalb des Programms wieder in den Normalzustand versetzt.

```
10 POKE 23659,0
20 FOR A=0 TO 23
30 PRINT A
40 NEXT A
50 PAUSE 0
60 POKE 23659,2
```

Um zu demonstrieren, was schief gehen kann, fügen wir diese Zeile in das Programm ein, um einen Fehler zu erzeugen:

```
45 PRINT error
```

Ups!!! Wenn Sie nur in die unteren beiden Zeilen ausgeben wollen, ist es in der Regel besser, **PRINT #1; "text"** zu verwenden, was genauso gut, wenn nicht sogar besser funktioniert, ohne das Risiko, einen Systemabsturz zu verursachen. Wenn Sie mit **POKE** einen Wert größer als zwei in **DF_SZ** eingeben, wird der obere Bildschirm kleiner als normal. Nach **POKE 23569,Y** wäre der obere Bildschirm also 24-Y Zeilen tiefer und würde scrollen, wenn die PRINT-Position 24-Y,0 erreicht oder überschreitet.

Dieses Programm zeigt, wie ein Teil des Bildschirms mit **DF_SZ** und **SCR_CT** gescrollt werden kann. Hier erscheinen Zufallszahlen, die nur in den oberen 14 Zeilen des Bildschirms nach oben blättern.

```
10 POKE 23692,0: POKE 23659,10
20 PRINT RND
30 GO TO 10
```

23670/1 SEED
NÄCHSTE ZUFALLSZAHL

Wenn **RANDOMIZE(Zahl)** verwendet wird, wird die Zahl (eine Konstante oder eine Variable) in dieser Systemvariablen gespeichert. Dies ist die Zahl, die die nächste Zufallszahl bestimmt. Dies eröffnet die Möglichkeit des Betrugs, da man die nächste (vermeintlich) generierte Zufallszahl ausrechnen und das gewonnene Wissen nutzen kann, um das Glück in die eigene Richtung zu "schwingen". Nach **RANDOMIZE 1** wäre der nächste Wert von **RND** beispielsweise **(0,0022735596, INT (RND *6) + 1)**, um einen Würfelwurf mit einer Eins zu simulieren.

23672/3/4 FRAMES
BILDERZÄHLER

Dies ist ein Bildzähler, der als Timer verwendet werden kann. Er zählt die Bilder eines Fernsehbildes und wird daher fünfzigmal pro Sekunde oder alle 0,02 Sekunden inkrementiert, obwohl die Zeit, die benötigt wird, um diese drei Bytes des Timers tatsächlich zu lesen und auszuwerten, es möglicherweise nicht erlaubt, ihn mit dieser Genauigkeit zu verwenden. Er hat einen Zeitbereich von fast vier Tagen (eigentlich etwa drei Tage 21¾ Stunden). Im Handbuch (Kapitel 18) wird darauf hingewiesen, dass man den Wert dieser drei Bytes zweimal hintereinander ablesen und den höchsten Wert nehmen muss, um die volle Genauigkeit zu erreichen, da sich die Werte der drei Bytes beim Ablesen so verändern können, dass große Ungenauigkeiten entstehen.

Es muss betont werden, dass die Timer-Bytes in der umgekehrten Reihenfolge stehen, als man erwarten würde - das höchstwertige Byte ist 23674, also werden die Timer-Werte gelesen durch:

```
65636*PEEK 23674+256*
PEEK 23673+PEEK 23672
```

was die Zeit in Einheiten von Fünfzigstelsekunden zurückgibt.

Es gibt mehrere Faktoren, die die Genauigkeit dieses Zeitmessers beeinflussen. Die Verwendung von **BEEP** hält den Timer an. Die Verwendung des Druckers und das Laden/Speichern usw. beeinträchtigen ebenfalls die Genauigkeit des Timers. Die Verwendung von **PAUSE** ist jedoch in Ordnung, da damit nur eine bestimmte Zeit gewartet wird, ohne dass der Timer neu eingestellt oder angehalten wird.

23675/6 UDG
BEGINN DER UDG'S

Die Adresse für den Beginn der Punktmuster für die benutzerdefinierten Grafiken ist normalerweise 32600 auf einem 16K Spectrum oder 65368 auf einem 48K Spectrum. Diese Zahl ist die gleiche wie **USR "a"**, also entspricht **PRINT USR "a":**

```
PRINT PEEK 23675+256*PEEK 23676
```

Zwanghafte **POKE**-Spieler können damit ihren Spaß haben. Das Handbuch schlägt vor, dies zu

ändern, um Platz zu sparen und weniger benutzerdefinierte Grafiken zu haben. Es ist jedoch auch umgekehrt möglich, mehr als einen Satz benutzerdefinierter Grafiken einzurichten, wenn dies erforderlich ist; es kann jedoch immer nur ein Satz von 21 verwendet werden. Da es **21 UDGs** gibt, ist es notwendig, **21*8 (168) Bytes** für jeden einzelnen Satz von **UDGs** zu reservieren und die Startadressen des verwendeten Zeichensatzes in 23675/6 einzutragen (**POKE**).

Geben Sie zum Spaß die folgenden Befehle ein:

```
POKE 23675,96: POKE 23676,127
(16K Spektrum)

POKE 23675,96: POKE 23676,255
(48K Spectrum)
```

Versuchen Sie dann, mit Hilfe der benutzerdefinierten Grafiken (sie erscheinen normalerweise als Großbuchstaben, bis sie neu definiert werden) eine Nachricht zu schreiben. Ich überlasse es Ihnen, herauszufinden, was passiert.

Ein nützlicher Tipp: Sobald Sie einen benutzerdefinierten Zeichensatz eingerichtet haben, können Sie diesen auf Band speichern. Die meisten Leute würden etwas eingeben wie:

```
SAVE "chars" CODE 32600,168
```

Gut, aber Sie müssen die Startadressen angeben. Sie könnten

```
SAVE "chars" CODE(PEEK 23675+256*
PEEK 23676),168
```

verwenden, und dann könnten Sie den aktuellen Satz von UDGs auf Band speichern, ohne die Adresse zu kennen, an der sie beginnen. Auf diese Weise könnte man z.B. einen Zeichensatz, der von einem 16K-Spectrum gespeichert wurde, in einen 48K-Spectrum zurückladen, ohne die Adressen kennen zu müssen.

Um einen Zeichensatz auf einem Rechner mit unterschiedlichem Speicherplatz wieder an die richtige Stelle zu bringen, verwenden Sie einfach

```
LOAD "chars" CODE(PEEK 23675+256*
PEEK 23676),168.
```

Dadurch werden die Daten automatisch an die richtige Adresse für den gerade verwendeten Rechner verschoben. Dies ist dasselbe wie

```
LOAD "chars" CODE USR "a",
```

was ein wenig Tipparbeit erspart, auch wenn es ein wenig seltsam aussehen mag.

23679 P_POSN
LPRINT-POSITION

Enthält Informationen darüber, wie weit die **LPRINT**-Position im Druckerpuffer fortgeschritten ist. Enthält (33- Spaltennummer) für die Spalten 0 bis 31.

23680 PR_CC
LOWBYTE NÄCHSTES ZEICHEN IM DRUCKERPUFFER

Enthält das niederwertige Byte der Adresse, an der das nächste Zeichen in den Druckerpuffer eingegeben ist, d.h. es enthält **(23296 + LPRINT**-Spaltennummer), wobei Null für die linke Spalte des Druckers steht, 15 für die 15te Spalte usw. Da dies die Adresse der obersten Punktreihe jedes Zeichens ist, können Sie mit **POKE** die Position des LPRINT-Puffers ändern, sofern Sie den Wert in **P_POSN (23679**) entsprechend ändern. Es scheint zu funktionieren, wenn Sie dies nicht tun, aber am Ende der Zeile treten Probleme auf.

23681
UNBENUTZTE SYSTEMVARIABLE

Diese Systemvariable, die streng genommen nicht verwendet wird, enthält normalerweise den Wert *91*. Dies ist das High-Byte der **LPRINT**-Pufferadresse (**91*256** ist **23296**, wo der Puffer beginnt). Sie kann für den eigenen Gebrauch ge**POKE**t werden, wird aber bei Benutzung des Druckers wieder auf 91 zurückgeschrieben. 23680/1 enthält die Adresse der **LPRINT**-Position im Druckerpuffer. Sie werden die Arbeit des Druckers nicht beeinflussen, wenn Sie 23681 **POKE**en, aber alles, was hier gespeichert ist, kann von den Druckerroutinen überschrieben werden.

23677/8 COORDS
X-KOORDINATE VON PLOT

23677 ist die Systemvariable, die die **x**-Koordinate des letzten gezeichneten Punktes enthält. Nach **CLS** beginnt sie bei Null und **23678** ist die Systemvariable, die die **y**-Koordinate des letzten gezeichneten Punktes enthält. Sie enthält den aktuellen Wert, d. h. wenn der letzte gezeichnete Punkt 3,3 war, enthalten beide Bytes Drei'en.

Diese beiden können mit gültigen **x**- bzw. **y**-Koordinaten ge**POKE**t werden. Da durch **POKE d** nichts auf dem Bildschirm geplottet wird, ist dies ein bequemer Weg, den **PLOT**-Cursor zu bewegen. Dies könnte durch **PLOT OVER 1;X,Y: PLOT OVER 1;X,Y** geschehen - wäre aber unübersichtlich. Unter anderem könnte dies **MOVE** simulieren, das man in anderen BASICs findet - nützlich, wenn man Linien um einen bestimmten Punkt herum zeichnen möchte.

23684/5 DF_CC
ADRESSE DER PRINTPOSITION

Die Adresse der **PRINT**-Position in der Anzeigedatei. Es kann **POKE d** sein, um die **PRINT**-Ausgabe an eine andere Stelle zu senden, obwohl dies ein Verständnis der Art und Weise erfordert, wie die Anzeigedatei organisiert ist.

23688/9 SPOSN
PRINTPOSITION AUßERHABB DER ZEILENBREITE

23688 enthält Informationen darüber, wie weit die **PRINT**-Position über den Bildschirm hinausgegangen ist. Sie beginnt mit 33 für die linke Seite des Bildschirms und nimmt jedes Mal um eins ab, wenn sich die **PRINT**-Position um eine Stelle nach rechts bewegt. Nach der Verwendung von **PRINT AT Y,X;** (wenn **Y** und **X** gültige **PRINT AT**-Koordinaten sind) würde 23688 33-X enthalten.

Dies kann nützlich sein, wenn man verhindern will, dass Wörter beim Ausdruck auf dem Bildschirm in zwei Hälften zerhackt werden. Wenn man sich vorstellt, dass die Zahl in 23688 gegen Null zählt, da in der aktuellen Zeile kein Platz mehr ist, kann man sehen, dass ein Vergleich mit der Länge des zu druckenden Wortes Aufschluss darüber gibt, ob es notwendig ist, in eine neue Zeile zu wechseln, um zu verhindern, dass das Wort zerhackt wird Angenommen, das zu druckende Wort wäre W$:

```
IF PEEK 23688<LEN W$+1 THEN PRINT
```

Dies funktioniert nur bei Wörtern, die weniger als 32 Zeichen lang sind.

23689 enthält Informationen darüber, wie weit unten auf dem Bildschirm die Druckposition erreicht wurde. Sie beginnt bei 24 für die oberste Zeile des Bildschirms und verringert sich jedes Mal um eins, wenn die **PRINT**-Position auf dem Bildschirm nach unten wandert. Wenn Sie keine scrollende Anzeige wünschen und lieber möchten, dass der Bildschirm gelöscht wird, wenn sich die **PRINT**-Position dem unteren Rand des Bildschirms nähert, dann versuchen Sie es:

```
IF PEEK 23689=3 THEN CLS
```

23693 ATTR_P
GLOBALE ATTRIBUTE

Enthält permanente Attribute oder die Attribute (**FLASH**, **BRIGHT**, **PAPER** und **INK**), die global gelten. Lokale Farben in **PRINT**-Anweisungen usw. werden an anderer Stelle behandelt. Beachten Sie, dass die meisten ROM-Routinen die Werte der Systemvariablen verwenden, die die temporären Attribute enthalten, da diese die permanenten Attribute enthalten, sofern kein lokaler Parameter angegeben ist. **CLS** hingegen stellt den Bildschirm auf die Farben usw. in **ATTR_P** ein. Die Funktionen der einzelnen Bits sind in Tabelle 1 dargestellt.

```
Bit 7 ist eins für FLASH 1.
Bit 7 ist Null für FLASH 0.
Bit 6 ist eins für BRIGHT 1.
Bit 6 ist Null für BRIGHT 0.
```

Die Bits 5, 4 und 3 enthalten die **PAPER**-Farbe in binärer Form, z. B. für **PAPER 7** wären die **Bits 5, 4** und **3** jeweils **1, 1** und **1**.

Bit 2, 1 und **0** enthalten die Farbe **INK** in binärer Form, z. B. für **INK 3** wären die **Bits 2, 1** und **0** jeweils **0, 1** und **1**.

Attribute von acht oder neun werden hier nicht behandelt. Wenn die permanenten Attribute acht oder neun sind, sind die in 23693 gespeicherten möglicherweise nicht gültig.

23694 MASK_P
ATTRIBUTE GEDRUCKTE ELEMENTE

Dies ist die Systemvariable, mit deren Hilfe Spectrum die Attribute aller gedruckten Elemente bestimmt, wenn ein Parameter von acht angegeben wird. Wenn Sie also **BRIGHT 8** global angegeben haben, wird Bit 6 von 23694 auf 1 gesetzt, um den Computer in Zukunft daran zu erinnern, dass **BRIGHT 8** angegeben wurde. Um die **Farbe**/ das **Blinken**/ die **Helligkeit** beim Drucken zu bestimmen, schaut der Computer also, was bereits vorhanden ist und druckt das Wort in dieser Farbe usw. Oder er überdruckt nur das Zeichen auf dem Bildschirm und lässt die Attribute in Ruhe. Was die einzelnen Bits bewirken, können Sie in Tabelle 2 nachlesen.

- **Bit 7 ist eins, wenn FLASH 8 in Kraft ist.**
- **Bit 6 ist eins, wenn BRIGHT 8 in Kraft ist.**
- **Die Bits 5, 4 und 3 sind normalerweise alle eins, wenn PAPER 8 aktiv ist, aber siehe unten.**
- **Die Bits 2, 1 und 0 sind normalerweise alle 1, wenn INK 8 wirksam ist, aber siehe unten.**

Wenn mehr als ein Bit zu berücksichtigen ist, wie bei **INK** und **PAPER**, dann werden nur für die gesetzten Bits die Attributbits vom Bildschirm genommen. Dies kann zu einigen unerwarteten Effekten führen. Versuchen Sie dies:

```
10 INK 8
20 POKE 23694, BIN 00000011
30 PRINT AT 0,0; INK 5;"5555555"
40 PRINT AT 0,0; "1111"
```

Da **INK 8** angegeben ist, könnte man erwarten, dass die **Einsen** wie die **Fünfen** in **Cyan** gedruckt werden, aber nein. Anstatt das **INK**-Attribut als Ganzes zu prüfen, werden nur die Bits geprüft, die in 23694 gesetzt sind, nämlich die Bits 0 und 1. Versuchen Sie herauszufinden, in welcher Farbe die **Einsen** gedruckt werden. Viel Spaß dabei!

23695 ATTR_T
AKTUELLE TEMPORÄRE FARBEN

Diese Systemvariable enthält die aktuellen temporären Farben, wie sie durch lokale Anweisungen innerhalb von **PRINT**-Anweisungen eingerichtet werden. Sie können sich das mit diesen beiden direkten Befehlen selbst ansehen:

```
PRINT PEEK 23695
PRINT INK 7; PAPER 0; PEEK 23695
```

Das heißt, Sie fügen das **PEEK** in eine **PRINT**-Anweisung ein, die die Wirkung der lokalen Farbsteuerungen hat. Wenn keine lokalen Farbanweisungen angegeben sind, enthält diese Systemvariable normalerweise die globalen Farbwerte. Die für den Druck auf dem Bildschirm zu verwendenden Farben usw. werden aus diesen temporären Systemvariablen entnommen, und die Dinge werden so ausbalanciert, dass sich **ATTR_T** nur dann von **ATTR_P** unterscheidet, wenn lokale Farbattribute usw. dies vorschreiben. Die Funktion der einzelnen Bits ist in Tabelle 3 zu sehen.

23696 MASK_T
ATTRIBUTE TEMPORÄRE ELEMENTE

Dies ist ähnlich wie **MASK_P** (Systemvariable 23694), nur dass die Parameter hier temporär sind. Normalerweise ist dies derselbe wie der entsprechende permanente Parameter 8s, der geändert wird, während die lokalen Farben 8s usw. in Kraft sind. Sie können dies untersuchen, indem Sie etwas wie folgt verwenden:

```
PRINT PEEK 23696, INK 8; PEEK 23696,
INK 0; FLASH 8; PEEK 23696
```

Die einzelnen Bits haben die in Tabelle 4 dargestellten Funktionen.

23697 P_FLAG
DRUCK-FLAGS

Diese Systemvariable enthält, wie ihr Name vermuten lässt, Flags, die beim Drucken verwendet werden. Nachdem **PAPER 9** angegeben wurde, werden die Bits 6 und 7 auf eins gesetzt. Nach der Angabe von **INK 9** werden die Bits 4 und 5 auf 1 gesetzt. Nachdem **INVERSE 1** angegeben wurde, werden die Bits 2 und 3 auf Eins gesetzt. Und nach der Angabe von **OVER 1** werden die Bits 0 und 1 auf Eins gesetzt. Die Auswirkungen sind global, wenn die ungeraden Bits (Bits 1, 3, 5 und 7) auf 1 gesetzt werden, und temporär, wenn die geraden Bits (Bits 0, 2, 4 und 6) auf 1 gesetzt werden. Siehe Tabelle 5, um zu sehen, was ich meine.

23681 23728/9
NICHT VERWENDET

Diese drei Bytes in den Systemvariablen werden normalerweise nicht vom Spectrum verwendet - Sie können sie als »benutzerdefinierte Variablen« für Ihre eigenen Programme verwenden, in denen Sie auf Informationen zugreifen müssen. Diese sind besonders nützlich in Maschinencode-Routinen, wo Sie einfach über eine Adresse auf die Informationen zugreifen können, anstatt die Variable im Variablenbereich zu suchen. 23728/9 war für die Verwendung durch nicht maskierbare Interrupts vorgesehen, die auf dem Spectrum jedoch nicht vorkommen.

23730/1 RAMTOP
LETZTE BYTE DES BASIC-SPIECHERS

Diese Zwei-Byte-Systemvariable verweist auf das letzte Byte des RAM des Basic-Systembereichs. Beachten Sie, dass dies nicht das Ende des von Basic verwendeten Speichers ist, da sich die benutzerdefinierten Grafiken normalerweise oberhalb dieser Adresse befinden. Wenn Sie RAMTOP über den Anfang der benutzerdefinierten Grafiken verschieben, werden diese möglicherweise überschrieben, aber Sie gewinnen einige wertvolle Bytes, die für 16K-Benutzer nützlich sein können.

Ein wichtiger Punkt ist, dass **NEW** nur bis zu der Adresse in 23730/1 funktioniert, so dass Sie darüber hinaus Daten speichern können, die *zwischen* in den Computer geladenen Programmen weitergegeben werden können. Das Gleiche gilt, wenn Sie Maschinencode-Routinen usw. aufbewahren wollen.

23732/3 P_RAMT

Hier steht die Adresse, an der das RAM des Spectrum's endet. Wenn Sie einen Spectrum erwerben, dessen Speicherkapazität Sie nicht kennen, brauchen Sie nicht nachzuschauen, ob es sich um ein erweitertes Modell handelt oder nicht, geben Sie einfach diesen Ausdruck ein:

```
PRINT PEEK 23732+256*
PEEK 23733-16384
```

Die 16384 Bytes, die abgezogen werden, sind für das ROM, da das RAM an der Adresse 16384 beginnt und bis zu der in **P_RAMT** enthaltenen Adresse reicht.

Die nebenstehenden Tabellen werden im Haupttext des ***zweiten Teils*** des Artikels referenziert und zeigen die Funktionen der einzelnen Bits der angegebenen Systemvariablen:

Tabelle 1: 23693 ATTR_P

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FLASH	*BRIGHT*	*PAPER colour binär*			*INK colour binär*		

Tabelle 2: *23694 MASK_P*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	
temp. FLASH	*temp. BRIGHT*	*temp. PAPER colour*			*temp. INK colour*		

Tabelle 3: *23695 ATTR_T*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
temp. FLASH8?	*temp. BRIGHT8?*	*temp. PAPER8?*			*temp. INK8?*		

Tabelle 4: *23696 MASK_T*

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
FLASH8	*BRIGHT8*	*PAPER colour 8?*			*INK colour 8?*		

Tabelle 5: *23697 P_FLAG*							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT1**
global PAPER9	*temp. PAPER9*	*global INK 9*	*temp. NK9*	*global INVERSE1*	*temp. INVERSE1*	*global OVER1*	*temp. OVER1*

Dieser Artikel ist ein Auszug aus dem Buch von DILWYN JONES, DELVING DEEPER INTO YOUR SPECTRUM ROM, das erstmals in Großbritannien von INTERFACE PUBLICATIONS veröffentlicht wurde und £7,95 kostete.

Die Links zu den Original-Artikeln lauten wie folgt:

YOUR SPECTRUM, Issue 2,
MARCH 1984 - SYSTEM VARIABLES

`http://www.users.globalnet.co.uk/~jg27paw4/yr02/yr02_78.htm`

YOUR SPECTRUM, ISSUE 3,
MAY 1984 - SYSTEM VARIABLES

`http://www.users.globalnet.co.uk/~jg27paw4/yr03/yr03_97.htm`

ZUM SCHLUSS:

Hier, unabhängig von dem vorherigen Artikel noch ein paar nützliche **POKEs**:

- Gesamten Speicherplatz ermitteln:
  `PRINT PEEK 23732 + PEEK 23733 * 256`
  `16kB-Spectrum: 32767`
  `48kB-Spectrum: 65535`
- Aktuellen freien Speicherplatz während des Programmlaufes ermitteln:
  `PRINT 65535-(PEEK 23653+256*PEEK 23654`
- Scroll bis 255 Seiten:
  `POKE 23692,255` oder
  `POKE 23692,0`
- Tastenton:
  `POKE 23609,50`
- Cursorgeschwindigkeit:
  `POKE 23562,1`
- Zeitmessung:
  `10 POKE 23672,0: POKE 23673,0`
  `20 LET S=(PEEK 23672+256*`
  `PEEK 23673)/50`
- Erste Basiczeile auf **0** setzen:
  `LET a=PEEK23637+256*PEEK 23638:`
  `POKE a,0: POKE a+1,0: POKE 23756,x`
  *Setzt die erste Zeile im Programm auf Zeilennummer **x**. Wenn x=0 ist, kann die erste Zeile nicht mehr einfach editiert werden. Der POKE-Befehl kann natürlich auch dazu benutzt werden, das Ganze rückgängig zu machen.*
- RAMTOP ermitteln:
  `PRINT PEEK 23730+256*PEEK 23731`

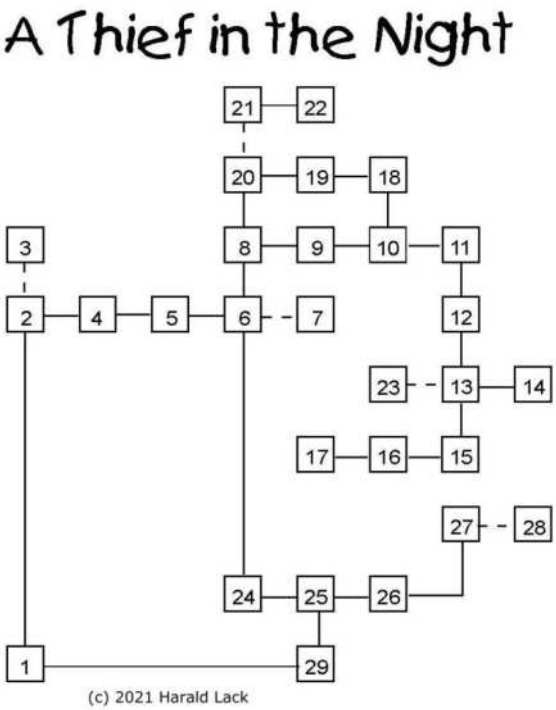
Fertige Routinen im ROM:

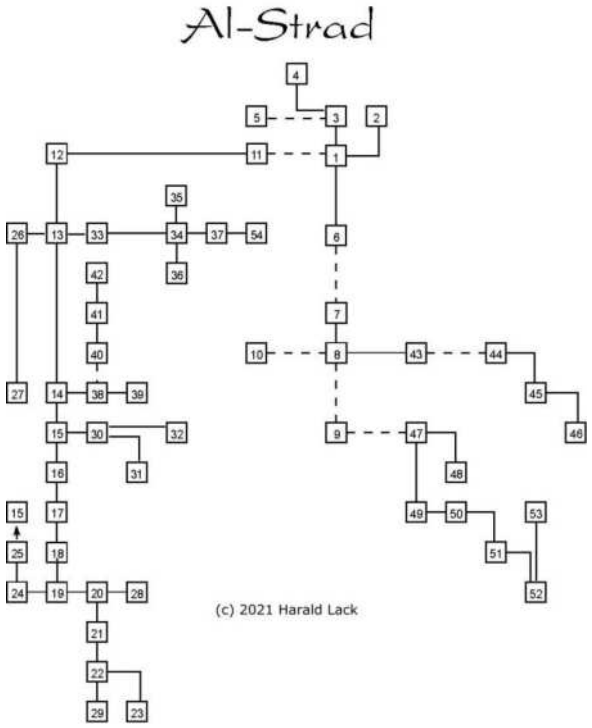
```
RANDOMIZE USR 3435:   CLS
RANDOMIZE USR 3756:   COPY
RANDOMIZE USR 6137:   LIST ab Zeile 0
RANDOMIZE USR 3438:   Löscht Zeile 23 & 24
```

NACHTRAG ZU DEN HEFTEN 229 & 230

Leider sind die Pläne zur Adventurelösung von ***Al-STRAD*** aus Heft 229 und ***A Thief in the Night*** aus Heft 230 unterwegs verloren gegangen – ich hab sie einfach vergessen. Man möge mir verzeihen!

Hier sind die Pläne!





DUAL-ROM BEIM ZX SPECTRUM 128K+2

VON FRANK DOPIERALA

Mein letzter Artikel endete zwar damit, dass ich noch auf die Bauteile zur Instandsetzung meiner Microdrive-Laufwerke wartete, in der Zwischenzeit hatte mich jedoch das Fertigstellungsfieber einer anderen »Baustelle« erwischt: Mein **ZX SPECTRUM 128K+2**-Restaurationsprojekt.

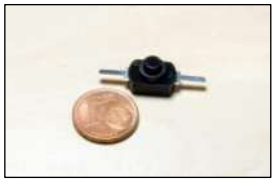
Im Grunde war er ja fertig restauriert und doch noch nicht fertig (ROM-Umschaltung). Das lies mir keine Ruhe, also habe ich kurzerhand dieses Projekt dem der Instandsetzung der Microdrives vorgezogen. Die Teile für diese sind auch mittlerweile alle da und es läuft mir ja nicht weg ☺.

Wenn ich mich recht entsinne, hatte ich beim letzten Mal erwähnt, dass bei solchen Retro-Arbeiten für mich nicht nur das Ergebnis zählt, sondern auch der Weg - ich habe keine Eile! Und da in mir ein pingeliger Perfektionist innewohnt, dauern kleine Detailarbeiten, die manch einer als Überflüssig ansehen könnte, bei mir eine Ewigkeit, bis sie fertiggestellt sind. Und fertig sind sie erst dann, wenn der Perfektionist in mir sein OK gegeben hat. Eine der Eigenschaften meines Perfektionisten beim Umbau oder technischen Ergänzungen aller Art besteht darin, dass der Umbau möglichst nicht nerdig-Techie-gebastelt aussehen soll, sondern nach Möglichkeit so, als hätte es der Hersteller selbst so vorgesehen. Technische Eleganz ist mir wichtig.

Das Original ROM in meinem **ZX SPECTRUM 128K+2** ist glücklicherweise gesockelt. Ansonsten hätte ich hier vielleicht verführt werden können, einfach ein weiteres EPROM huckepack auf das bereits eingelötet ROM draufzupacken, wie ich es ganz früher bei meinen ersten Speccys auch gemacht habe 😇. Ich hatte auch überlegt, ob ich diese Technik trotzdem anwende oder ob ich ein doppelt so großes EPROM nehme, um darin beide ROM-Images, also das vom +2 und das vom **128K TOASTRACK**, zu brennen. Hierzu musste ich erst mal prüfen, inwieweit ein 27256 und ein 27512 EPROM pinkompatibel sind. Glücklicherweise sind sie es, lediglich der PIN für die Programmierspannung liegt woanders, aber das erkennt der JUNIOR-PROMMER automatisch. Somit war für mich klar, dass ich zu einem einzigen 64K-EPROM greifen werde, um beide unterzubringen. Allerdings wusste ich in diesem Moment noch nicht, wie ich beide Images zu einer Datei zusammengefasst brennen konnte. Auch hier half mir das EPROM-Brennprogramm **PINATUBO** für meinen **ATARI TT** ganz intuitiv weiter: Ich gebe das 27512 EPROM als das zu brennende Medium an und lade dann als zu verwendendes Image einfach eines der beiden 32K-Dateien. Daraufhin »meckert« **PINATUBO**, dass das Image zu klein ist für das ausgewählte EPROM. Hier kann ich aber unter »Datei« nun »Anfügen« anstatt »Laden« auswählen und schon hängt **PINATUBO** automatisch die zweiten 32K hinten dran - fertig.

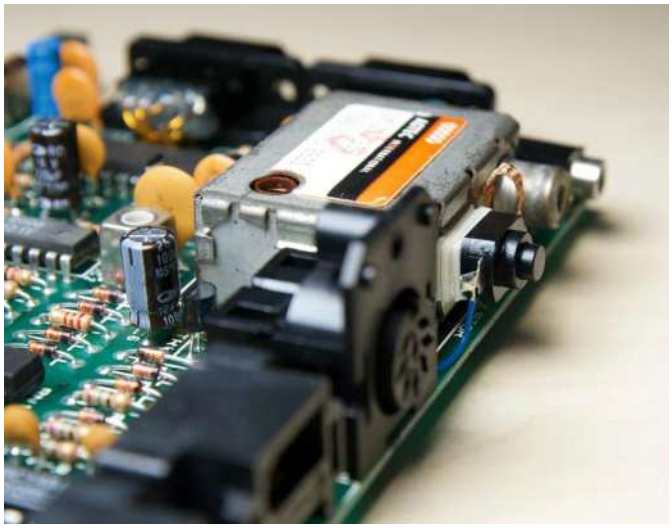
Nachdem das EPROM gebrannt wurde, habe ich es erst einmal 1:1 gegen das Original ROM getauscht. PIN 1 des Sockels im Spectrum ist direkt mit +5V versehen, somit musste das »angehängte« ROM-Image aktiv sein. Eingeschaltet: Alles klar! Meldet sich mit der Amstrad-Meldung. Danach die Diag-Karte drangehangen und von dieser die Prüfsumme checken lassen: Alles OK! Also, Spectrum aus, EPROM raus, PIN 1 vom EPROM etwas rausgebogen und mit Krokoklemme direkt mit Masse verbunden: Sinclair Meldung!! Nochmal mit der Diag-Karte gecheckt, alles Top! Jetzt konnte ich das EPROM mit einem entsprechenden Aufkleber beschriften, als UV-Schutz und damit es hübsch aussieht.

Als nächstes stellte ich mir die Frage, wie ich die Umschaltung konzipieren sollte. Zunächst dachte ich an einen kleinen Schiebeschalter an der rechten Seite vom Spectrum, quasi gegenüber der Joystick-Ports. Dazu hätte ich aber mindestens zwei Kabel zum Mainboard führen müssen. Und entweder ich lötete diese Kabel dann direkt an oder ich musste mir noch eine Steckverbindung überlegen, wenn ich das Mainboard mal ausbauen wollte. Beides gefiel mir nicht. Also habe ich mir das Mainboard und das Gehäuseunterteil genauer angeschaut. Da gibt es diesen kleinen Spalt zwischen Gehäuse und Modulator. Ich googelte nach Miniatur-Schaltern und wurde bei Ebay fündig. Bestellt, gemessen und begeistert.



Der Schalter war sogar so schön rechteckig, dass er genau zwischen Bodenplatte und Deckel des Modulators passt. Nachdem ich die Position des Schalters mit eingesetzten Mainboard genau am Gehäuse angezeichnet hatte, wurde das Loch gebohrt. Den Schalter habe ich dann mit vier Lagen doppelseitigem und elastischem Klebeband an den Modulator geklebt. Die vier Lagen haben sich durch experimentieren herauskristallisiert: Es mussten gerade so viele Lagen sein, dass der Schalter beim Einsetzen des Mainboards mit ausreichendem Druck gegen die Gehäusewand hält. Auch dann, wenn der Schalter betätigt wird. Danach die beiden Lötfahnen ge-

kürzt und an einer der beiden ein Stück Entlötlitze gegen die Modulator-Basisplatte gelötet, damit der Schalter beim Einsetzen des Mainboards auch weiterhin beweglich bleibt.



Da der Schalter lediglich Masse durchschaltet, muss das EPROM an PIN 1 (Adressleitung A15 auf Masse = Low = Adressbereich 0 bis 32767, A15 auf +5V = High = Adressbereich 32768 bis 65535) also durch einen Pull-Up Widerstand auf +5V gesetzt werden, falls der Schalter geöffnet ist. Da direkt gegenüber an PIN 28 die Versorgungsspannung für das EPROM anliegt, eine saubere Sache: Einfach einen 1K Pull-Up Widerstand quer aufgelötet, fertig! Vorab habe ich ein Stück Schaltdraht genau auf die Länge gekürzt, die ich für die Verbindung zwischen Schalter und EPROM-PIN benötige. Diese

habe ich dann vorab im 45 Grad-Winkel nach unten an PIN 1 des EPROM gelötet. Das andere Ende des Schaltdrahtes habe ich dann abwärts Richtung Mainboard an die zweite Lötfahne des Schalters gelötet. Den Verlauf des Drahtes auf dem Mainboard kann man auf dem Bild sehen.

Wenn das Mainboard nun eingesetzt wird, findet der Schalter quasi automatisch seine Führung mit dem Loch in der Rückwand und passt sich perfekt an. Der Schalter ist so konzipiert, dass er etwas herausteht, wenn er geschlossen ist und nahezu plan mit dem Gehäuse ist, wenn er schließt. Nun noch der abschließende Funktionstest, zunächst mit geschlossenem (herausstehendem) Schalter. Spectrum ausschalten, Schalter eindrücken, nochmal einschalten: Voila! Die Technik und die Umsetzung des Schalters gefallen mir soweit sehr gut!

Aber, da fehlt doch noch das kleine Quäntchen, bis mein Inneres Ruhe gibt - die fehlende Beschriftung lässt den Schalter als nachträglichen Einbau erkennen. Das muss geändert werden und zwar mit Buchstabenbeschriftungsfolie, in diesem Fall durch Buchstaben, die sich durch Wasser von der Trägerfolie ablösen und dann gut positioniert auftragen lassen. Erst nachdem ich die zweizeilige Beschriftung angebracht habe, gefällt mir das Gesamtergebnis. JETZT ist mein Restaurationsprojekt „**ZX Spectrum 128K +2**“ abgeschlossen und ich kann endlich eines meiner Lieblingsspiele mit dem Speccy spielen, die da sind **»Cyclone«, »Highway Encounter«, »Worse Things Happen at Sea«** ...

SHARP MZ 700 UND MZ 800

VON JOACHIM GEUPEL

Heute möchte ich euch zwei weitere Computer Klassiker aus meiner Sammlung vorstellen. Beide habe ich, als sie auf den Markt gekommen sind, im Rahmen meiner Tätigkeit in einem Computerladen selbst erlebt. Es sind die Computer ***MZ-700*** und ***MZ-800*** von **Sharp**. Beide sind im Aufbau sehr ähnlich und in der Ausstattung fast identisch – bis auf ein paar grundlegenden Merkmale.

Die Computer hätten wegweisend sein können, sind jedoch in der Versenkung verschwunden, wie so viele andere, die Anfang der 80er herausgebracht wurden.

HAYAKAWA METALS

Viele werden es vermuten, andere tatsächlich wissen: **Sharp** hat einmal Wasserhähne gebaut, Gürtelschnallen für Gürtel ohne Löcher, und natürlich auch mechanische Bleistifte. Mal ganz ehrlich?! Wer hat das nicht gewusst?

Sharp wurde 1912 von Herr ***Tokuji Hayakawa*** gegründet, kurz nachdem er die besagte Gürtelschnalle erfunden und patentieren ließ. Herr

HAYAKAWA war ein recht genialer Erfinder, sozusagen der japanische Clive Sinclair – nur war er auf längere Sicht viel erfolgreicher. Die Werkstatt, die er gründete, nannte sich **HAYAKAWA METALS**. Sein mechanischer Bleistift, der **HAYAKAWA MECHANICAL PENCIL**, war ein großer Erfolg und wurde über Japan hinaus auch im Ausland patentiert. In welcher Stadt das Vorläuferunternehmen von **SHARP** gegründet wurde, ist nicht ganz klar. Was aber klar ist, ist der Firmenmotto, dass *Herr Hayakawa* seinen Produkten zugrunde gelegt hat. Er meinte *»Make products, that other companies want to imitate«*, was so viel heißt wie *»baue Produkte, die andere nachahmen wollten«*, was wohl seinen Erfolg begründet.

Nach dem *Großen Kantō-Erdbeben*, das große Teile von Tokio und die Hafenstadt Yokohama zerstörte, siedelte das Unternehmen nach Osaka um. 1925 entstand dort der erste Kristall-Detektor, einfachste Geräte zum Empfang von Rundfunksendungen, denen 1929 das erste Röhrenradio folgte. Ab 1931 befasste sich die Firma mit der Erforschung und Entwicklung von Fernsehgeräten und war damit eine der ersten Firmen weltweit, die den funktechnische Empfang von Übertragungen bewegter Bildern in Angriff nahm. 1935 ging das Unternehmen an die Börse und 1942 nannte sich um in **HAYAKAWA ELECTRIC**.

SHARP

Ab 1953 baute das Unternehmen als erstes japanisches Unternehmen Fernsehgeräte in Serie. Das erste Fernsehgerät von **SHARP**, das **TV3-14T**, kostete 175.000 Yen - damals lag das Anfangsgehalt für Schulabgänger bei 5.400 Yen im Monat. Das Gerät hat somit locker 32 Monatsgehälter gekostet. Etwa zur gleichen Zeit änderte **HAYAKAWA ELECTRIC** seinen Firmennamen in **SHARP**. Der Name leitete sich aus der Bezeichnung des populären mechanischen Bleistift ab, den *EVER-READY* **SHARP PENCIL,** kurz **EVER-SHARP PENCIL** genannt.

Schon recht früh, nämlich 1963 entwickelte Sharp den ersten transistorbasierten Tischrechner **SHARP COMPET CS-10A,** ein Tischrechner mit Nixieröhren-Anzeige, der 1964 heraus kam.

Die Entwicklung von Rechnern und Computern zieht sich durch bis in die späten 80er-Jahre. 1978 brachte **SHARP** den **MZ80K** heraus, einer der ersten echten Desktop-Computer, die frei programmierbar waren. Der **MZ80K** stand in direkter Konkurrenz zum **APPLE II,** dem *TRS-80* von **RADIO SHACK** und dem **PET** von **COMMODORE**. Etwa zeitgleich zum **MZ80A** kam der **SHARP MZ 3500** heraus, der eher für den kommerziellen Bereich gedacht war und einem *IBM-PC* recht ähnlich sah. Dem **MZ80K** folgte der **MZ80A** und als dessen direkter Nachfolger der **MZ-700**.

Parallel zu den Tisch- und Home-Computern machte sich Sharp einen Namen mit den Pocket-Computern. Was heute ein Personal Computer, also ein PC ist, war in den 80ern ebenfalls ein PC, was aber für **POCKET COMPUTER** stand. 1983 wurde der Taschen- oder Pocket Computer **PC 1245** unter das Volk gebracht, ein Jahr später folgte der **PC1246**. Zwei dieser Pocket Computer befinden sich ein meiner Sammlung, die ich irgendwann später vorstellen werde. Beendet wurde die Pocket-Computer-Reihe mit dem **SHARP PC 2500**, wobei dieser eher einer der frühen Laptops war. *Der* **HC-4100** war im Jahr 1997 schon ein Handheld-Computer mit LCD-Display und **WINDOWS CE 2.0**.

Meine persönlichen Erinnerungen an die beiden Computer beziehen sich auf die Zeit Ende 1984 und Anfang 1985. Als ich in den letzten beiden letzten Monaten von 1984 in einem Computerladen in Reutlingen jobbte, waren die beiden Computer in diesem Laden echte Stars. Der Chef des Ladens war klein und dick, hatte zwar Ahnung von Geld aber nur rudimentäre Ahnung von Computern. Der Laden war vollgestellt mit allen möglichen Arten von Computern. Neben **EPSON QX-10** Computern, die mit **CP/M** liefen, standen auch noch **ORIC'S**, der obligatorische **C64**, die ersten **CPC-COMPUTER** und ein oder zwei **ZX SPECTRUM** herum. Seinen Hauptumsatz machte er mit den **ALPHATRONIC PC'S**, bei denen er die 5 ¼-Zoll Diskettenlaufwerke durch die damals hypermodernen 3 1/2 –Zoll Diskettenlaufwerke ersetzen ließ. Während der **TA-PC8** relativ unspektakulär in einer Ecke aufgebaut war, standen die beiden **SHARP**-Computer als Blickfang im Eingangsbereich. Jeder, der den Laden betrat, stolperte zwangsläufig über die Computer. Während der **MZ-800** mit farbiger, hochauflösender Grafik geglänzt hat, mussten am **MZ-731** alle zwei Tage die Zeichenstifte des Plotters ersetzt werden, weil er, sobald Kunden in meinem Alter und die etwas Jüngeren den Plotter gerne in Beschlag nahmen, um die Demo-Grafiken und eigene Kunstwerke zu Papier brachten. Alle Computer nutzen ihm nichts. Ein Jahr später war er pleite und ging später für ein Jahr wegen Unterschlagung und mehrfachen betrügerischen Konkurses in den Knast. Mich selbst hat das nicht mehr berührt. Ich hatte zum Jahreswechsel 84/85 eine Virusgrippe und lag mit Fieber im Bett. Während ich krank war, hat er mich rausgeschmissen. Ich fand einen anderen Job in einer Elektronikfirma, und genau ein Jahr nach meiner Entlassung habe ich an der Zwangsversteigerung des restlichen Inventars teilgenommen…

DER MZ-700

Der **SHARP MZ-700** ist, ebenso wie sein Vorläufer **MZ80K** und sein Nachfolger **MZ-800** ein sogenannter **CLEAN COMPUTER**. Darunter verstand **SHARP**, dass den Computern über nur ein 4kB großes ROM ver-

fügte, das einen Assembler-Monitor, Ein/Ausgabe-Routinen für den Monitor und den Lautsprecher und eine Lade-Routine für den Kassettenrekorder beinhaltete. Das obligatorische Basic wurde von Kassette eingeladen, was den Nachteil hatte, dass man mehrere Minuten warten musste, bis es gestartete. Der Vorteil war, dass jede verfügbare Programmiersprache geladen werden konnte. Es gab eine Menge an Sprachen, die für diese Maschine angepasst waren. So waren mehr als fünf Versionen von BASIC verfügbar, außerdem mehrere *Assembler*, *Pascal*, *Lisp*, *C*, *Fortran*, *Comal* (eine Synthese aus *Basic* und *Pacal*), *Forth* und einige andere.

Der Computer wurde in vier verschiedenen Ausbaustufen ausgeliefert. Das Basismodel war der **MZ-711**. Er verfügte nur über die notwendigste Grundausstattung. Grundsätzlich ist es möglich, einen RGB-Monitor anzuschießen, ebenso wie einen normalen Video-Monitor und das obligatorischen Fernsehgerät. Ein Centronics-kompatibles Druckerport, dessen Anschluss ähnlich dem Edge Connector der **SINCLAIR ZX-COMPUTER** nach außen geführt wurde, ist ebenfalls vorhanden. Zwei 3,5 mm Klinkenbuchsen ermöglichen das Laden und Speichern von Programmen. Zwei Joystick-Ports, die als fünfpolige Stiftleisten herausgeführt sind, sind zu nichts kompatibel. Es ist nicht ohne Aufwand möglich, einen Atari-Kompatiblen Joystick anzuschließen.

Der **MZ-721** verfügt über einen eingebauten Kassettenrekorder, der auf der rechten Seite eingebaut ist. Die Ear- und Mic-Buchse auf der Rückseite werden durch den Einbau des Kassettenrekorders nich ersetzt, können aber nichtmehr benutzt werden.

Der **MZ-731** hat mittig einen Vierfarb-Plotter eingebaut, über den sowohl Grafiken und Zeichnungen geplottet werden können, als auch Listings ausgegeben werden können, was natürlich seine Zeit benötigt.

In alle Modelle kann anstelle des Kassettenlaufwerks ein **QUICK-DRIVE** eingesetzt werden. Das **QUICK-DRIVE** ist eine Diskettenlaufwerk für 2,8-Zoll-Disketten. Sie sind mit nichts kompatibel. Statt in Spuren, wie es bei anderen Diskettenformaten üblich ist, werden die Disketten in einer durchgehenden Spirale beschrieben, ähnlich einer Schallplatte. Sowohl das **QUICK-DRIVE** als auch die 2,8-Zoll-Disketten sind in der Zwischenzeit schwer zu bekommen – und wenn, dann sind sie exorbitant teuer.

Das Motherboard des **MZ-700** macht einen sehr aufgeräumten Eindruck. Alles ist solide verarbeitet und übersichtlich im Gehäuse untergebracht. Der Lautsprecher hat einen Durchmesser von 77mm und macht ordentlich Lärm, das Netzteil ist ein kompaktes Schaltnetzteil und der Videomodulator stellt neben dem obligatorischen HF-Ausgang einen Compositvideo-Anschluss und eine RGB-Buchse zur Verfügung.

SHARP hat dem Computer einen Customer-Chip spendiert. Sowohl der Speicher-Controller als auch der CRT-Controller sind in einem einzigen kundenspezifischen LSI-Chip, dem M60719 enthalten. Er vereint auf sich dem 8x8-Dot-Generator, der 40 Zeichen und 25 Zeilen auf den Bildschirm ausgibt. Die angezeigte Schriftart hängt vom 4-kB-ROM ab, in dem die Zeichen abgelegt sind. Er verwaltet das Monitor-ROM, das DRAM, das Video-RAM und die Peripheriegeräte Tastatur, Timer, usw. , die dem Speicher zugeordnet sind. Er erzeugt den Takt für den Z80A-Prozessor und wählt den Drucker-E/A-Anschluss aus. Dieses IC ist klein aber fein.

Der Prozessor ist ein offenbar von **SHARP** in Lizenz gebauter Z80A mit der Bezeichnung LH0080A. Damit kein Zweifel aufkommt, ist die Prozessorbezeichnung daneben aufgedruckt.

Auf der Platine befinden sich zwei unübersehbare Schiebeschalter. Mit dem einen wird zwischen eingebautem und externem Kassettenrekorder umgeschaltet. Der zweite Schalter selektiert zwischen eingebautem Plotter und externem Drucker.

Der Computer wurde für ungefähr 800 DM angeboten, wobei die Preisangaben, die im Internet zu finden sind, stark schwanken. Auf einer Seite stehen 248.000 Yen, was ungefähr 3000 DM entspricht, auf einer anderen Seite sind es 510 DM. Ich meine zu wissen, dass der Computerladen ihn für besagte 800 DM verkauft hat.

DER SHARP MZ-800

Wie der **MZ-700** *ist* auch der **MZ-800** ein **CLEAN COMPUTER**. Er ist der Nachfolger der **MZ-700**-Reihe. Er ist ähnlich ausgestattet wie der **MZ-700**, kann

aber auf 128 kB aufgerüstet werden.

SHARP hat auch hier die einzelnen Modelle je nach Ausstattung aufgeteilt. Auf den ersten Eindruck wirkt er etwas wuchtiger als sein Vorgänger. Das Grundgerät, der **MZ-811**, verfügt weder über ein Kassettenlaufwerk noch über einen Plotter und in der die Version **MZ-821** ist noch ein Kassettenlaufwerk integriert. Die Anschlüsse sind die gleichen wie im **MZ-700***:* HF- und Composit-Ausgang, RGB, Joysticks und ein Centronics-Port.

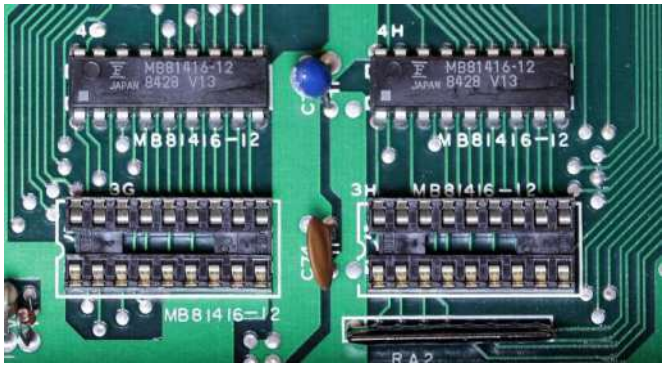
Hinter einer Abdeckung verbergen sich zwei Steckplätze für Erweiterungen.

Der Computer unterscheidet sich schon äußerlich von seinem Vorgänger. Das Oberteil des Gehäuses ist in einem durchgängigen cremefarbenen Weiß, die Tasten, und die im **MZ-700** sich in einem mehr oder weniger schönen Ocker zeigen, sind im **MZ-800** in einem dunklen Grau.

Während im **MZ-700** der hintere Teil des Gehäuseoberteils noch dreigeteilt ist, um das Kassettenlaufwerk und den Plotter aufzunehmen, ist im **MZ-800** kein interner Plotter mehr vorgesehen. Den Platz nehmen stattdessen die zwei Steckplätze ein, in die Interfaces wie z.B. für ein Diskettenlaufwerk oder eine RAM-Disk eingesteckt werden können.

Auch im **MZ-800** arbeitet einZ80-Prozessor. Im **MZ-700** wird er mit 3,5MHz getaktet, im **MZ-800** mit 3,55MHz. In der Textdarstellung begnügte sich der **MZ-700** noch mit 40 Zeichen in 25 Zeilen, der **MZ-800** hat zwei Textmodi, nämlich bei gleicher Zeilenanzahl umschaltbar von 40 auf 80 Zeichen pro Zeile.

Grafisch ist der MZ-800 besser ausgestattet. Während der MZ-700 nur sieben Farben hat, können beim MZ-800 16 Farbe dargestellt werden. Die Anzahl der Pixel ist beim MZ-800 mit 320 x 200 bzw. 640 x 200 um einiges größer als bei seinem kleinen Bruder, der nur 80 x 50 Grafikpunkte in entsprechend großen Blöcken hat.



Die Hauptplatine ist wie die des MZ-700 schön

aufgeräumt und übersichtlich. Auch hier findet sich ein Sharp-eigenes LSI-Chip zur Speicherverwaltung und CRT-Controller. Augenfällig ist der breite Stecker, in den das Flachbandkabel für die beiden Erweiterungsplätze gesteckt wird.

Auf der Platine sind zwei leere IC-Sockel zu finden, in die noch zwei 16kB x 4 Bit-Speicher-IC's gesteckt werden können. Es ist anzunehmen, dass das Monitor-Ram aufgerüstet werden kann, indem entsprechende ICs hineingesteckt werden.

In diversen Beschreibungen des Computers ist zu lesen, dass er auf 128 kB aufgerüstete werden kann. Steckplätze für RAMs sind, bis auf die besagten IC-Sockel auf der Platine nicht vorhanden. Eine zweite Annahme von mir ist, dass die Speichererweiterung auf 128kB in Form einer Platine erfolgt, die in eine der beiden freien Steckplätze geschoben wird.

Alles in allem ist der MZ-800 ein sehr schöner Computer. Er ist sehr gut verarbeitet, lässt sich leicht demontieren und wieder zusammenbauen und sieht einfach gut aus. Es ist von Vorteil, wenn ein Kassettenlaufwerk eingebaut ist. Ohne diesen Datenrekorder ist es schwierig, Programme zu laden, weil der Computer nicht mit jedem externen Rekorder arbeiten will und herum zickt.

FAZIT

Beide Computer machen Spaß, wenn sie funktionieren. Sie sind solide verarbeitet und sehen gut aus. Sind allerdings die Sharp-eigene LSI-IC defekt, eignen sie sich nur noch als dekoratives Element in einer Ausstellung mit dem Verweis, dass das einmal ein Computer war.

jg

ADVENTURELÖSUNG »CALLING«
VON HARALD LACK

The Calling (of the Demon)

1987 Terry Taylor and F. J. Neary (Visual Dimensions)

Liebe Mituser!

Man glaubt es kaum, was einem so alles in die Hände fällt wenn man seinen ganzen Fundus mal gezielt durchsucht. Bei mir ist das oben genannte Adventure aufgetaucht, das sich sicher schon zu lange in meinen dunklen Kisten versteckt hat. Es wurde 1987 von TERRY TAYLOR und F. J. NEARY unter Verwendung des Professional ADVENTURE WRITERS geschrieben und von VISUAL DIMENSIONS veröffentlicht. Das Programm gibt es sowohl als 48er als auch 128er Version, wobei sich meine Lösung auf die Version für den SPECTRUM 48K bezieht. Ob es inhaltliche Unterschiede bei den beiden Versionen gibt ist mir nicht bekannt.

Das Programm bietet die Möglichkeit RAM SAVE und RAM LOAD zu benutzen wie auch Bilder ein oder aus zuschalten. Das macht die Sache dann auch ein bisschen praktischer, man muss nicht so viel mit externen Speichermedien hantieren.

Also angepackt und durchgespielt habe ich mir gedacht... was gar nicht so einfach ist und mich einige Monate grübeln gekostet hat.

Das Programm an sich gehört für mich grundsätzlich zu den guten Adventures, die Richtungen stimmen und auch die Locations-Beschreibungen geben was her. Was etwas nervt, auch wenn es letztendlich insgesamt nicht übermäßig stört, ist die Quengelei des Programmes, wenn man sich mal was in Ruhe durchlesen will. Dadurch, dass man immer wieder zur Aktivität aufgefordert wird, verschwindet manchmal der Text der Location zu früh, ehe man in ganz durchgelesen hat, was ja bei Adventures durchaus Sinn machen kann, denn nicht selten sind dort hilfreiche Tipps und Hinweise versteckt. Im Zweifelsfall halt einfach nochmal *»look«* eingeben und der Text ist wieder da.

Auffällig ist, dass das Programm jede Menge Gegenstände enthält, die aber zur Lösung selten beitragen. Das war bestimmt eine kleine Gemeinheit der Programmierer.

Das größte Hindernis bei der Lösung stellt meiner Meinung nach die Stelle dar, wo wir uns in unserem Studierzimmer befinden und den CHARM benutzt haben, denn danach ist ein SAY CONFIDENCE angesagt, das sich einem nicht unbedingt sofort erschließt (siehe Tipps weiter unten). Soviel vorneweg.

Zur Abrundung jetzt noch die kurze Hintergrundgeschichte zum Programm: Der Spieler befindet sich mit seiner Freundin *Jenny* auf der Rückfahrt von York, wo wir bei einem Abendessen waren, über das *Yorkshire Moor* in Richtung *Bayley*, unserem Zuhause, als uns mitten Überland unser gutes Auto im Stich lässt. Zudem herrscht noch ein Unwetter und ein Sturm kommt auf. Naja, sowas soll es tatsächlich geben. Sofort bietet *Jenny* an Hilfe zu holen und ist auch schon weg ehe wir uns versehen.

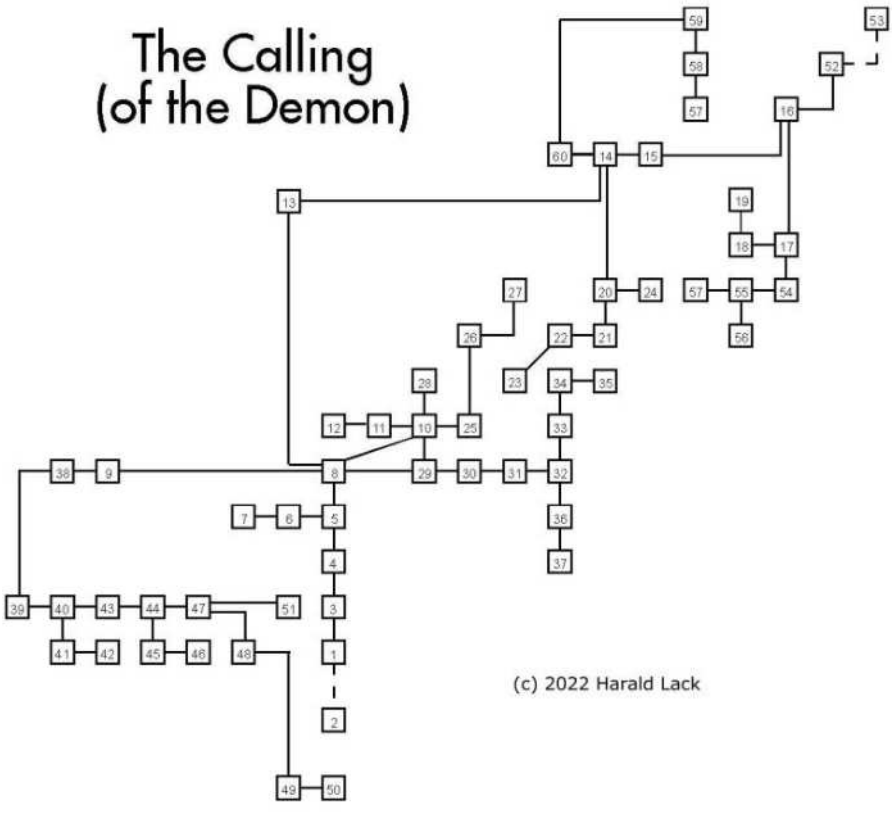
In einem nahegelegenen Haus versucht sie den Abschleppdienst zu verständigen. Leider gibt es danach von ihr keine Lebenszeichen mehr, so dass wir uns nach zweistündigem Warten auf die Suche nach ihr begeben, da wir vermuten, dass ihr etwas zugestoßen ist, zumal sich dieses merkwürdiges Anwesen hier allein auf weiter Flur befindet. Außerdem benötigen wir ja noch den Abschleppwagen, da sich unser Gefährt sicher nicht mehr von selbst von der Stelle bewegen wird. Somit wissen wir grundsätzlich mal, worin unsere Aufgabe bestehen wird.

Ich bin mir natürlich bewusst, dass viele Spieler ein Adventure gerne selbst lösen wollen. Deshalb vor meiner schrittweisen Lösung (ich hoffe ich habe meine handschriftlichen Aufzeichnungen immer richtig übertragen was bei meiner Schrift oft schon ein eigenes Abenteuer ist) zuerst noch vier Tipps, die eventuell weiterhelfen auch wenn man die Gesamtlösung nicht benutzen will.

- Man kann zur selben Zeit immer nur einen Schlüssel tragen. Da es deren zwei (RUSTY KEY und SKELETON KEY) im Programm gibt, ist es sinnvoll sie an der richtigen Stelle abzulegen.
- In diesem Adventure ist es durchaus sinnvoll die eine oder andere Person anzusprechen, um vielleicht wertvolle Hinweise zu bekommen. Der Parser versteht auch relativ komplexe Sätze. Wichtig ist es jedoch, dass man beim Ansprechen Anführungszeichen (" ") benutzt und zwar beispielsweise in der Form: Say to x "follow me".
- Im Laufe des Adventures werden wir auf einen nützlichen Kupferhelm stoßen. Dieser wird uns gute Dienste leisten. Jedoch ist unbedingt darauf zu achten, diesen **nicht außerhalb** des Gebäudes zu tragen, da er Blitze anzieht, was uns nicht gut bekommt.
- Wenn wir *Jenny* gefunden haben, wird uns diese folgen. Leider ist sie etwas geschwächt und hängt uns hinterher. Deshalb ist es ganz besonders am Ende unbedingt wichtig, auf sie zu warten, da ihr Telefonat zur Lösung des Spieles wesentlich beiträgt.

Soviel zu den vier Tipps, die ich an den Anfang stellen möchte. Somit kann sich jeder selbst auf den Lösungsweg begeben. Nachfolgenden nun die insgesamt 60 Locations meines Planes mit seinen darin befindlichen Gegenständen (soweit im Rahmen meiner Lösung ersichtlich bzw. erforderlich):

01) Standing in an old country lane
02) Within the broken down red car / pair of gloves
03) Walking along a dark pavel driveway
04) Standing outside a large imposing house
05) Standing in the middle of a vast entrance hall / some armour
06) In a large dusty room
07) In a spacious music room / grand piano, flute
08) Walking along the main corridor
09) In the old study / oak desk, diary, hole in the west wall
10) In the kitchen of the house / kitchen table, carving knife
11) Walking in an east-west corridor
12) In the drably decorated servants quarters / music magazine
13) Standing upon the highly ornate main staircase
14) Standing on the rickety landing
15) Standing in the base room of the north-east tower
16) Standing upon the cold stone floor of the north-easttower´s lower room
17) Standing under a sloping roof at the east end of the large attic
18) In the middle of the large attic room / picture on the north wall
19) In the previously hidden room / rusty key
20) Striding along the upper corridor A
21) Striding along the upper corridor B
22) Within the walls of a highly decorated bedroom
23) In a long low room / metal sphere
24) Standing in the upper gallery / valuable painting, skeleton key
25) In the neatly arranged kitchen garden / some garlick, some hemlock
26) Standing under the branches of a tree
27) Perched among the damp leaves of the tree / crystal ring
28) Among the smartly scrubbed shelves of the pantry / cooked ham
29) Entering the dining hall of the house
30) Standing on the damp paving stones of the terrace
31) In the middle of the beautiful gardens
32) Strinding along the damp gravel path that runs north-south
33) Standing upon a gravel path
34) Standing in a large grassy hollow
35) Inside a large kennel-like building / some junk, golden charm
36) On a gravel path in the rain-soaked garden
37) Standing within the confines of the gardener´s lodge / sturdy chest, piece of paper
38) In a gloomy alcove
39) At the bottom of the spiral stairs
40) At the west end of the long cellar corridor
41) In the low-roofed living quarters / pair if overalls
42) Standing in a small but well equiped kitchen / chocolate biscuit
43) Walking along the east-west cellar corridor
44) Strolling along the east-west corridor that runs the full length of the cellar
45) Standing in Quinn´s strange laboratory
46) In the corner of the cluttered laboratory / copper helmet
47) At the east end of the cellar corridor
48) Walking along the dark flight of stairs
49) Deep in the bowels of the earth
50) Standing below the ceiling of a vast subterranean tomb / some robes, large tomb
51) Standing in the dimly lit robing chamber / some boots
52) Standing at the top of the north-east tower
53) Clinging pecariously to a platform / strange machine, Quinn, Jenny
54) In the attic library
55) Walking along an attic corridor
56) In a small but used attic study / small desk, telephone
57) Stading in a part of the attic that had been made into a store
58) Standing under the sloping roof to the west of the large attic
59) Within the confines of the north-west tower´s lower room
60) Standing in the base room of the north-west tower



Das war alles zum Plan und den Gegenständen. Für alle jene, die es (noch) nicht selbst versuchen wollen geht es jetzt so weiter...

(Wir starten in a country lane mit heftigem Unwetter in einem kaputten Auto), INVENTORY (wir tragen einen Anzug - wir kommen ja vom Essen), ENTER CAR, GET GLOVES, EXIT CAR, N, N (nun ausserhalb des besagten Hauses), N (die Eingangshalle), EXAMINE ARMOUR, W, W (das Musikzimmer - hoffentlich gut schallisoliert), GET FLUTE, PLAY FLUTE (wir benötigen zuerst noch eine Musikzeitung), E, E, N, W (das Studierzimmer), READ DIARY (es gehört Professor Quinn), E, NE (die Küche), W, W (der Bereich der Dienerschaft), GET music MAGAZINE, EXAMINE music MAGAZINE, E, E, SW, U (Treppenaufgang), U (auf dem Treppenabsatz), E, U, S, W (die Mitte des Dachgeschosses), EXAMINE PICTURE (stellt ein menschliches Ohr dar), PLAY FLUTE (durch unser ungeschicktes Flöte spielen verändert sich das Bild, löst sich schließlich auf und gibt einen bisher verborgenen Gang frei), N (der unbedingt notwendige Geheimraum - was wären Adventures ohne ihn?), GET RUSTY KEY (die darin enthaltene Magie verbrennt uns die Finger und wir lassen ihn lieber fallen), WEAR GLOVES, GET RUSTY KEY (schon besser), S, DROP FLUTE, DROP MAGAZINE, E, N, D (am Fundament des Nordostturms), W, S, S, W (ein Schlafzimmer), OPEN DOOR, SW, GET SPHERE, EXAMINE SPHERE (diese stellt künftig unsere Lichtquelle dar), NE, E, N, N, D, D, DROP RUSTY KEY (das mit den Schlüsseln habe ich oben schon erwähnt - es kann nur einer sein - Highlander Prinzip), U, U, S, E (die obere Gallerie), EXAMINE valuable PAINTING (ein alter Mann, der ein Pferd reitet), EXAMINE PAINTING CAREFULLY (da ist der zweite Schlüssel dieses Adventures), GET skeleton KEY, EXAMINE KEY, W, N, D, D, NE (wieder die Küche), N (der Küchenchef will uns nicht in die Vorratskammer lassen), E (der Gemüsegarten der Küche), GET GARLIC, GET HEMLOCK, EXAMINE HEMLOCK (giftig!!!), N (am Fusse eines Baumes), U (auf dem Baum), GET crystal RING, EXAMINE RING, D, S, W (zurück zur Küche), GIVE GARLIC TO COOK (genau darauf wahr der Küchenchef fixiert), N (die Vorratskammer), GET HAM, EXAMINE HAM (wir entdecken einen kleinen Einschnitt an der Seite, der von einem Messer stammen könnte), PUT HEMLOCK IN HAM (und schon ist der Schinken vergiftet), S (zurück in die Küche), GET KNIFE, EXAMINE KNIFE (überraschend scharf), S, E (auf die Terrasse), E (und weiter in den Garten), E (ein Kiesweg), N, N (eine grasige Senke), E (im Rinnstein, hier ist ein schuppiges Biest das nach dem Schinken giert; da er vergiftet ist, tut ihm das nicht gut und es löst sich nach und nach auf - kommt davon, wenn man so gierig ist), EXAMINE JUNK CAREFULLY (wir finden einen goldenen Zaubergegenstand), GET CHARM, EXAMINE CHARM (darauf befinden sich magische Symbole),W, S, S, S, S (wir erreichen die Hütte des Gärtners, der über unser Erscheinen nicht sehr glücklich ist, denn er schaut sehr böse - an dieser Stelle können wir nur eine Aktion tätigen, also keine Zeit verlieren), KILL GARDENER WITH KNIFE, DROP KNIFE, EXAMINE CHEST (hat keinen Deckel), LOOK IN CHEST, GET PAPER FROM CHEST, READ PAPER (beinhaltet einige Hinweise für uns), N, N, W, W, W, W (der Hauptgang),
W (wieder das Studierzimmer), EXAMINE OAK DESK, INSERT CHARM (in das vorhandene Loch in der Wand - verschmilzt mit der Wand und lässt diese in einwandfreiem Zustand zurück), SAY 'CONFIDENCE' (wie oben gesagt, keinesfalls die Anführungszeichen vergessen; es ist ein lautes, brechendes Geräusch zu vernehmen und in der Westwand erscheint ein Durchgang), W (im Alkoven), WEAR RING, SAY TO SPHERE 'LIGHT' (man muss unbedingt den Ring übergezogen haben sonst wird´s nichts - das Leuchten beginnt), D (der unvermeidliche Keller), E, S (der Wohnbereich), GET OVERALLS, EXAMINE OVERALLS, WEAR OVERALLS, N, E (zur Küche), GET BISCUIT, EXAMINE BISCUIT, DROP BISCUIT, W, N, E, E, UNLOCK STEEL DOOR WITH SKELETON KEY, S (Professor Quinn's Labor), E (eine interessante Ecke im Labor), GET HELMET, EXAMINE HELMET (nur für den „Hausgebrauch" geeignet - siehe meine Ausführungen weiter oben), WEAR HELMET, W, N, E, D, D (der Eingang zum Grabmahl wo uns ein sehr alter Geist zu attacklieren versucht, aber wir haben ja den Helm, der uns davor schützt), E (hinein ins Grabmal), GET ROBES, EXAMINE ROBES, W, U, U, E (ein unsichtbares Kraftfeld hindert uns am Fortkommen), WEAR ROBES, E (das ist die Robenkammer im Tempel von Danrath), GET BOOTS, EXAMINE BOOTS, W, W, W, W, W, U, DROP SPHERE, REMOVE HELMET, DROP HELMET, REMOVE ROBES, DROP ROBES, DROP PAPER, E, E, GET RUSTY KEY, U, U (am Treppenabsatz), E, U (der Nordostturm), U (zur Spitze des Turmes - wir sollten das Loch in der Decke nicht übersehen), WEAR BOOTS, JUMP (wir springen dank unserer Stiefel durch das Loch in der Decke und landen auf einer durchaus etwas merkwürdigen Plattform… Professor Quinn ist auch hier aber als er uns sieht erschrickt er, rutscht aus und fällt), EXAMINE MACHINE, UNLOCK MACHINE WITH RUSTY KEY (Jenny ist darin eingesperrt und wir können sie befreien), D (Jenny folgt uns, wenn auch etwas langsam), D, S, S, W, S, EXAMINE TELEPHONE, WAIT (wir warten bis Jenny da ist und den Abschleppdienst anruft - dann möchte sie zurück zum Auto), N, W, N, N, D, E, D (das zentrale Treppenhaus), D, S, S, S, S (an der Landstrasse), WAIT (bis endlich der Abschleppwagen kommt und uns mitnimmt). Und schon werden wir mit der finalen Meldung belohnt die da lautet:

CONGRATULATIONS
You are on your way home and have completed the game.
Welldone! You have scored 100%

Dem möchte auch ich nichts mehr hinzufügen. Mir hat es großen Spaß gemacht, mich durch das Adventure zu arbeiten. Ich hoffe ich konnte einigen wertvolle Tipps geben.

Spectrum Next Spiele-Review
»Gold Compilation«
von Thomas Eberle

Mit der sogenannten **»Gold Compilation«** erhielt ich drei kommerzielle Spiele für den **Spectrum Next**, die ich hier einmal vorstellen möchte. Den meisten sollten die Titel schon bekannt sein, da aber nicht jeder das aktuelle Geschehen verfolgt, hier eine Zusammenfassung:

Die **Compilation** besteht aus drei Titeln der Fa. **Software Amusement**. Diese Firma hat schon mehrere Titel veröffentlicht, auch für moderne Systeme wie Android oder Amazon Fire TV. Diese drei Titel sind gleichzeitig die ersten Titel der Firma für den **Spectrum Next** und nach starten der SD-Karte werden die drei Titel in einem Menü mit Sound angezeigt:

1. **Dungeonette**
2. **Deltastar Earth Defense**
3. **Montana Mike**

Dungeonette

Das ganze Spiel erinnert an **Atic Atac** und dies ist wohl kein Zufall.

Man steuert einen Ritter durch ein Labyrinth, auf der Suche nach Schlüsseln um Türen zu öffnen und Schätze zu finden. Der Ritter hat zwar ein Schwert, kann aber lediglich kleine Messer werfen. Die Gegner die sich im Labyrinth finden, werden durch mehrere Messertreffer zerstört, manche auch gar nicht. Wird man von einem Gegner berührt, kann man sich kurz nicht bewegen, was meist einen weiteren Treffer verursacht, und man bekommt Energie abgezogen. Durch Essen kann man die Energie wieder auffüllen.

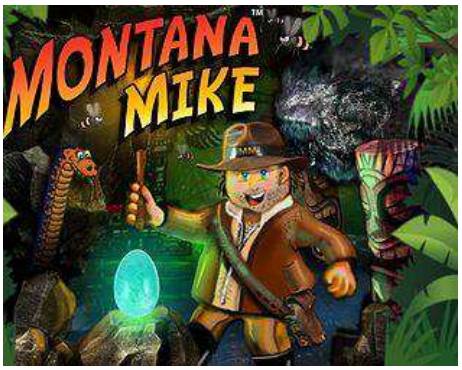
Ein Spiel das mal 1985 ein Erfolg gewesen ist, mag heute noch Erinnerungen hervorrufen, aber nicht wirklich begeistern. Auf Sounds verzichtet das Spiel völlig und die Grafik hebt sich nur durch die Farben, nicht aber durch Detailreichtum von einem durchschnittlichen Spectrum-Spiel ab. Insgesamt sicher kein Grund deswegen einen **NEXT** haben zu müssen.

Deltastar Earth Defense

Dieses Spiel ist eine Art **Galaxian**–Clone. Man steuert ein Raumschiff am unteren Bildschirmrand, während sich darüber Formationen von Aliens bilden, die sich teile im Sturzflug nach unten stürzen, dabei aus allen Rohren feuernd. Ziel des Spiels ist es alle Aliens abzuschießen, dann kommt der nächste Bildschirm. Für Abwechslung sorgen die unterschiedlichen Alien-Typen sowie einige Bonus-Schachteln, die auch von oben nach unten fallen. Man kann auf die Schachteln schießen und sie verändern die Wirkung die letztlich eintritt, wenn man sie einsammelt. Hier kann man neue Waffen, extra Leben oder Bonuspunkte bekommen.

Es ist nicht gerade innovativ, ein Spiel von 1979 neu aufzulegen. Zwar gibt es eine nette Weltraum-Hintergrundgrafik, aber ansonsten hat sich nicht viel getan. Im Spiel gibt es keine Musik, lediglich im Titelmenü begrüßt uns ein Soundtrack, der mich an meine eigenen ersten Versuche mit AY-Musik erinnert. Trotzdem durchaus spaßig kurz mal zwischendurch zu spielen.

Montana Mike



Wer sich noch an Rick Dangerous erinnert, findet hier einen Nachfolger für den NEXT. Wer es nicht kennt: Im Grunde ist ein ein Plattformer mit einem Held im Stile von Indiana Jones. So hat er aus Grundwaffe eine Peitsche, kann aber auch andere Waffen finden und nutzen. Man irrt durch eine Pyramide von Bildschirm zu Bildschirm, versucht in jedem Bild den Ausgang zu finden ohne dabei ein Leben zu verlieren. Der Held kann hüpfen, klettern und krabbeln, zumindest letzteres kommt nicht in jedem Plattformer vor.

Das Spiel hat eine ansehnliche Grafik, zumindest für Spectrum-Verhältnisse. Aus dem NEXT könnte man sicher mehr rausholen, schlecht ist es aber

nicht. Es gibt einen Titelsong im Hauptmenü, aber keine Musik im Spiel -Lediglich FX-Effekte.

Insgesamt ist die Compilation eine Umsetzung altbekannter Spielprinzipien auf ein neues Gerät. Immerhin: **SOFTWARE AMUSEMENT** war einer der ersten Semi-professionellen Vertriebe die sich des NEXTS angenommen haben, dies schon bevor die Geräte an die meisten ausgeliefert wurden. Die SD Karte mit den Spielen kommt in einer schönen Box , auf eine Anleitung der Spiele wird aber verzichtet. Leider liefen bei mir die Spiele erst überhaupt nicht, nach einlegen der SD Karte und booten erhielt ich ein wirres Bild. Erst nach einigen Nachforschungen fand ich heraus, dass dieses Spiel wohl den SCANDOUBLER aktiviert, da ich aber meinen Next über ein RGB-Kabel angeschlossen habe, kam das wirre Bild heraus. Kleiner Tipp hier: man drückt NMI und dann „2“ , das schaltet den Scandoubler wieder aus.

Der Verkauf der Spiele (und weitere von **SOFTWARE AMUSEMENT**) endete am 28.02.2021 und seitdem sind die Spiele frei verfügbar. Hier der Link:

https://itch.io/s/27215/zx-spectrum-next-games-bundle

GAMES 2019 / 2020

VON ELLVIS

So langsam nähern wir uns unserem Ziel, alle unserer Meinung nach sehr guten Programme vorgestellt zu haben bis zum aktuellen Zeitpunkt. Hier nun die Reviews bis zum Jahr 2020:

»AD LUNAM PLUS«
© 2019 / 2020 VON ALESSANDRO GRUSSU

Ich war nicht dabei, als die Menschheit zum allerersten Mal auf dem Mond landete, aber ich kann mir die Faszination eines ganzen Weltraumrennens in den 50er und 60er Jahren vorstellen. Und da die Computer der damaligen Zeit bei der Landung halfen, kann der Spectrum mir helfen, selbst das Weltraumrennen zu erleben und noch mehr... ein Teil davon zu sein!

ist eine erweiterte Version des rundenbasierten Strategiespiels über die Entwicklung der gesamten Ausrüstung und deren Einsatz zur Landung auf dem Mond. Das Originalspiel hatte ein paar Ecken und Kanten, aber die verbesserte Version ist es wert in der Freizeit gespielt zu werden.

Zu Beginn wählen wir das Land - USA oder UdSSR. Das Spiel bemüht sich darum, akkurat zu sein und das hilft, ein wenig Authentizität zu erreichen. Das erste Menü macht deutlich dass dieses Spiel eine Menge Optionen bietet und somit etwas ist, das uns für längere Zeit beschäftigen wird. Jede Wahl im Spiel wird durch die Auswahl einer Option mittels einer Zahl gemacht. Manchmal müssen wir einen Wert eingeben (z.B. die Anzahl der Forscher, die wir einem Projekt widmen wollen). Es gibt kein Zeitlimit für die Auswahl. Sobald wir eine Entscheidung getroffen haben, ist die letzte Option, unseren Zug zu beenden. Der Computer spielt das andere Land im Hintergrund und wir erhalten nach jeder beendeten Runde Berichte darüber. Dies geschieht in Form von Kurznachrichten, wie Berichte von Geheimdiensten, Änderungen in der Regierungsfinanzierung und so weiter. So bleiben wir im Bild und haben nicht das Gefühl, allein zu spielen. Eine Runde steht für 3 Monate, also haben wir 4 Runden pro Jahr.

Die Komplexität des Spiels ist ziemlich groß. Wir müssen ein Auge auf die Forschung und Entwicklung der Weltraumausrüstung (z.B. Wechsel der Forscher zwischen den Projekten), die Rekrutierung und Ausbildung der Astronauten, den Bau von Einrichtungen und auch die Planung von Missionen. Je weiter das Spiel fortschreitet, desto mehr Dinge müssen wir pro Zug beachten als zu Beginn des Spiels. Und genau darin liegt die Strategie, wir wollen doch die ersten Menschen auf dem Mond sein, oder?

Ziemlich oft werden wir Meldungen wie *»Mission gescheitert«* sehen. Lasst euch davon nicht abschrecken, es ist normal, dass die Dinge nicht von Anfang an perfekt laufen. Das ist Teil des Spiels und hilft bei der Entscheidung, worauf man sich konzentrieren sollte. Einfach ein paar Versuche mehr und du bist oben im All.

Das Spiel ist nur für 128K SPECTRUMS geeignet. Es enthält keine große Menge an Sound, aber es gibt eine wirklich schöne Grafik, die uns durch das Spiel führt. Das Konzept und die Steuerung sind sehr einfach und man braucht keine Zeit, um zu lernen, was was bedeutet und wie man spielt. Es gibt auch eine SAVE-Option, so dass wir das Spiel nicht von Anfang bis Ende durchspielen müssen, da dies einige Zeit in Anspruch nehmen würde. Wenn ihr Strategie ein bisschen mögt, solltet ihr **AD LUNAM PLUS** ausprobieren, ihr werdet nicht enttäuscht sein!

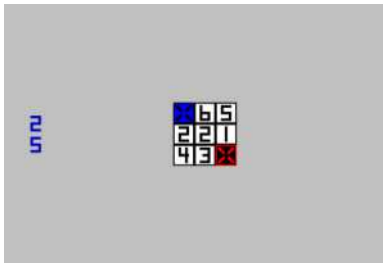
Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36138/ZX-Spectrum/Ad_Lunam_Plus

»CARDINAL CHAINS«
©2020 VON COMPILER-SOFTWARE (MIGUEL ANGEL GARCIA PRADA)

In der letzten Ausgabe haben wir **COLORISTIC**, ein kleines Logikspiel, genauer unter die Lupe genommen.

CARDINAL CHAINS ist im Grunde dasselbe mit einer kleinen Wendung. Wir müssen immer noch das gesamte Spielfeld mit einer oder mehreren Farben zu füllen (je nach Level). Der Unterschied ist, dass jede Zelle eine Zahl enthält. Wir können von einer



niedrigeren Zahl zur höheren Zahl gehen, aber wir können nicht umgekehrt von der höheren zur niedrigeren Zahl uns bewegen.

Und das ist der Hauptunterschied zwischen den beiden Spielen. Alles andere ist sehr ähnlich, wir haben ein minimales Bildschirmdesign und diesmal bekommen wir keine Musik.

Es gibt auch keine Passwörter, aber wir können mit Rechtes im Menü wählen, auf welcher Ebene das Spiel gestartet werden soll.

Aufgrund der Zahlen finde ich dieses Spiel etwas einfacher als **COLORISTIC**. Andererseits haben wir hier 313 Level vor uns, also bin ich mir sicher das mich das Spiel noch einige Zeit beschäftigen wird. Und ich bin sicher, wenn ihr **COLORISTIC** gemocht habt, werdet ihr dieses Spiel auch sehr spielenswert empfinden.

Download unter:

https://spectrumcomputing.co.uk/entry/36456/ZX-Spectrum/Cardinal_Chains

»COLOCO«
©2020 TUXEDO GAMES (MANU SEGURA)

Und wieder einmal sind wir auf einer Rettungsmission. Aber jetzt ohne Waffen und ohne Horden von Feinden. Klingt langweilig? Nun, das würde ich so sagen...

COLOCO ist eine interessante Abwandlung der Spiele vom Typ **„THRUST“.** Wir versuchen, Astronauten zu retten, die im Untergrund des Planeten und in den unterirdischen Einrichtungen herumirren. Da unsere Rakete klein ist, können wir nur eine Person pro Level retten.

Die ersten Level sehen einfach aus. Nur ein Bildschirm, wir starten auf der einen Seite und holen den Astronauten auf die andere Seite des Bildschirms und fliegen dann zurück. Aber sobald man gestartet ist, versteht man warum dieses Spiel nicht so einfach ist. Die Steuerung beachtet die Trägheit, so dass das Fliegen ziemlich schwierig ist. Es braucht ein paar Spiele, um sich sich daran zu gewöhnen. Aber sehen wir uns das Ganze mal genauer an. Es gibt 3 Arten von Plattformen auf denen wir landen können, gelb sind die, auf die wir die Astronauten bringen müssen, blau sind die auf denen die Astronauten sind, und grüne sind gelegentlich in den Levels verstreut, normalerweise mit einem hilfreichen Objekt darauf. Wir können nirgendwo anders landen als auf diesen Plattformen. Um die Sache ein wenig zu vereinfachen, kann man schon von ziemlich hoch oben beginnen zu landen, so dass wir bei der Landung selbst nicht allzu vorsichtig sein müssen.

Im Laufe der Levels müssen wir über mehrere Bildschirme fliegen. Auch Feinde tauchen auf, von Minen bis hin zu schwebenden Außerirdischen. Um die Dinge noch schwieriger zu machen werden die Astronauten hinter verschlossenen Türen festgehalten, so dass wir zuerst einen Schlüssel suchen müssen, aber das Herumfliegen kostet uns eine Menge Treibstoff. Glücklicherweise gibt es auch

Punkte mit Treibstoff zum Aufsammeln. Manchmal können wir sogar ein Bonusleben bekommen. Diese kann man bei jedem Spiel aufsammeln, so dass man am Ende in einigen Levels viele Versuche nutzen kann, oben am Ende wirklich weniger Leben zu haben. Zu Beginn des Spiels erhalten wir 15 Leben zur Verfügung, was viel erscheinen mag, aber spätere Levels werden uns zeigen das dem nicht so ist. Unter dem Spielfeld sehen wir ein Informationsfeld mit der Anzahl der Leben (links), aktuelle Ladung (rot, wenn unser Raumschiff leer ist, und grün, wenn wir einen Astronauten haben), Schlüssel (rot, wenn wir keinen haben und grün, wenn wir einen haben) einen) und schließlich den Treibstoffstatus.

Das Spiel verwendet die **CHURRERA-ENGINE MK1,** die aus vielen, vielen Plattformspielen bekannt ist, ich bin mir sicher, dass euch die Soundeffekten bekannt vorkommen werden, und dieses Spiel zeigt, wie vielseitig das Tool ist.

Die Grafik ist sehr schön und sauber, wir sehen immer, was vor sich geht. Die Sounds sind nur für den SPECTRUM 48K.

COLOCO ist ein interessantes Spiel und wenn ihr von all den Jump'n'Runs gelangweilt seid und eine andere Art von Action wollt, solltet ihr es ausprobieren. Es könnte euch sehr gut gefallen.

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36342/ZX-Spectrum/Coloco

»H.E.R.O. RETURNS«
© 2020 VON GUSMANB

Etwas älter zu sein bedeutet nicht immer nur ein Bier vor dem Fernseher zu trinken. Vor allem, wenn man in der Vergangenheit ein Held war und sieht, dass die Menschen einen noch brauchen.

Zeit, den Staub von deiner Ausrüstung zu schütteln und ein paar Leute zu retten.

H.E.R.O. war in den frühen 80er Jahren ein beliebtes Spiel. Das Spiel handelte von einem namenlosen Helden, der mit einem Fluganzug und ein paar Bomben ausgerüstet in die Höhlen geht und Menschen rettet. Das Spiel hat Spaß gemacht, und im Jahr 2020 haben wir nun eine Hommage an das Spiel.

Das Spiel beginnt an der Oberfläche mit einem Eingang zu einer Höhle. Sobald wir sie betreten haben, müssen wir das Ende der Höhle erreichen, wo die zu rettende Person sitzt. Das Gute daran ist dass der Weg in der Regel geradlinig ist und man sich nicht wirklich verirren kann. Um das zu kompensieren, macht das Spiel die Dinge schwieriger, indem es uns in enge Korridore führt, in denen wir weder Boden noch Decke berühren dürfen. Solche Passagen sind wirklich schwer zu passieren.

Die Höhlen sind nicht allzu groß, aber sie sind auch nicht leer. Ab und zu treffen wir auf eine Fledermaus, eine Spinne, eine Krabbe oder ein anderes Lebewesen. Einige von ihnen können mit unserer kleinen Laserpistole getötet werden. Wir haben auch Bomben, aber die haben keine Wirkung auf die Feinde und können nur zum Durchbrechen von Wänden verwendet werden. Neben den Feinden erwähnte ich bereits, dass einige Oberflächen von unserem Helden nicht berührt werden dürfen. Diese sind Lava - leuchtend roter/gelber Stein und Wasser. Wenn wir über das Wasser fliegen, müssen wir auf die Alligatoren aufpassen, die hin und wieder auftauchen.

Das Spiel sieht ganz anders aus als das Original. Die Grafik ist detaillierter, aber alles ist kleiner. Außerdem ist das Wasser nur eine blinkende Textur, aber das Original hatte schöne Wellen. Aber das Wichtigste ist das Gameplay und das ist immer noch ziemlich gut. Hilfreich ist hier, dass wir, sobald wir alle unsere Leben aufgebraucht haben (wir bekommen 4 beim Spielstart), mit weiteren 4 Leben an der Stelle auf dem Bildschirm weitermachen können, an der wir gestorben sind.

H.E.R.O. Returns ist ein nettes kleines Spiel, eine Hommage an die goldene Ära des Videospiels. Wenn du ein paar Minuten Zeit hast und ein bisschen frustrierendes Gameplay magst, solltest Du es ausprobieren.

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36387/ZX-Spectrum/HERO_Returns

»JUMPING JUPITER«
©2020 VON QUANTUM SHEEP

Ich werde nicht umhin kommen, mindestens einen gewöhnlichen Plattformer zu spielen. Meine Tasse Tee ist fast leer und dieses Spiel ist genau das, was ich brauche - das Sammeln von Teetassen überall im ganzen Sonnensystem. Also, los geht's!

Jumping Jupiter ist ein klassisches Plattformspiel in der Tradition von **Manic Minner**. Jedes der 35 Levels besteht nur aus einem Bildschirm, auf dem du alle Tassen und manchmal auch einen Schlüssel. Wenn das geschafft ist, betrittst du den Ausgang und gehst zum nächsten Level. Um es nicht zu einfach zu machen, gibt es Feinde, denen man ausweichen muss.

Das Schöne an dem Spiel ist, dass man die Hauptfigur (sie heißt *Jupa*) mit nur einer Hand steuern kann es genügen O, P und M. Die andere Hand bleibt aus dem Spiel und kann zum Beispiel eine Tasse Tee oder so halten. Einfacher wird die Sache auch wegen der Tatsache, dass wir von jeder Höhe fallen können, ohne ein Leben zu verlieren. Außerdem können wir bei den Sprüngen (und Stürzen) immer noch die Richtung kontrollieren und das ist in vielen Situationen eine große Hilfe. Obwohl die Geschwindigkeit der Hauptfigur nicht die höchste ist, trägt sie zur Action bei und die Spielbarkeit ist immer noch sehr hoch.

Ich fand die Levels eher einfach als schwierig, mit gelegentlichem Kampf hier und da. Es ist sehr einfach, ins Spiel einzusteigen und in kürzester Zeit Fortschritte zu machen. Die Spielbarkeit ist insgesamt sehr gut. Die Grafik ist schön, detailliert und einfach zu sehen, was vor sich geht. Der Sound ist wahrscheinlich der schlechteste Teil des Spieles. Es gibt überhaupt keine Soundeffekte, aber es läuft die ganze Zeit eine Melodie, die sehr bald auf die Nerven gehen kann. Die kosmische Stille wäre in diesem Fall besser würde ich sagen.

Alles in allem, schnappt euch dieses Spiel, wenn ihr noch Lust auf ein paar Sprünge über den Bildschirm.

Download unter:

https://quantumsheep.itch.io/jumpin-jupiter

»MARSMARE ALIENATION«
©2020 VON DRUNK FLY

Die Aliens haben es wieder getan. Sie haben dich mitten in der Nacht aus deinem Bett geholt. Da du geschlafen hast, war es eine leichte Aufgabe für sie. Aber sobald du aufgewacht bist, wirst du ihnen versichern, dass sie diesmal den Falschen erwischt haben.

Nach einer kurzen Einführung geht es auch schon los mit der Action. Unbewaffnet, in einer außerirdischen Raumstation, unsere Aufgabe ist es, nach Hause zu entkommen. Hinter dem Energiefeld befindet sich eine Waffe, aber der Weg dorthin ist nicht so einfach. Also gehen wir in eine andere Richtung um zu sehen, wie die Station aussieht.

Auf den meisten Bildschirmen sind Außerirdische zu sehen. Zuerst müssen wir ihnen ausweichen, da wir ohne eine Waffe nicht kämpfen können. Sobald wir eine haben, können wir sie erschießen. Eliminierte Feinde (Aliens, aber auch Roboter, Stromaggregate und so weiter) erscheinen nicht wieder wenn wir den Bildschirm erneut betreten. Es gibt auch eine Auto-Mapping-Funktion. Sobald wir die der Karte sehen, sehen wir auch wie viel

wir von unserem Ziel schon erreicht haben und wie viel mehr auf uns wartet.

Der Bahnhof ist voll funktionsfähig und das ist ein Vorteil für uns. Wir können Fahrstühle benutzen und das Wichtigste: es gibt Computerterminals, die wir benutzen können, um unsere Position zu speichern. Sobald wir sterben, sind wir wieder am letzten gespeicherten Punkt. Die Zerstörung von Energiegeräten ist ein Muss, sonst können wir die Energiefelder nicht passieren.

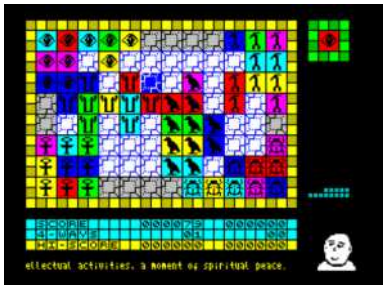
Wichtig sind auch die Energiekapseln - die brauchen wir für die Waffe und Pillen, die unsere Gesundheit wiederherstellen. Sobald wir einen Raumanzug haben, können wir die Station verlassen und die Umgebung erkunden.

Das Spiel selbst ist einfach wunderbar. Die Animationen sind sehr flüssig, die Grafik detailliert und sehr schön. Die Steuerung ist sehr präzise und die Spielbarkeit ist insgesamt einfach großartig und es gibt gute Musik im Hintergrund. Zusammengefasst ist Marsmare Alienation ein weiteres großartiges Spiel, ein Muss!

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36516/ZX-Spectrum/Marsmare_Alienation

»THE LEGACY OF THE WHITE CRANE - ISHIDO 2«
© FANZIX



ISHIDO war ein Remake eines alten Logik-Spiels, das 2015 auf dem Spectrum erschien. Nach weiteren 5 Jahren bekommen wir eine neue Version, die in einigen Punkten verbessert wurde. Wenn du ein Fan von Logikspielen bist, sollte deine Aufmerksamkeit in diese Richtung gehen.

Sobald du das Spiel geladen hast, wirst du von einem Mönch begrüßt. Er lebt in diesem Spiel und und er wird dich während des gesamten Spiels begleiten. Und das ist auch gut so denn er wird dir nicht nur zu Beginn die Geschichte erzählen und später die Regeln erklären, sondern er wird auch deine Züge während des Spiels kommentieren und dir sogar den nächsten Zug empfehlen. Es gibt auch eine Variante des Spiels, bei der der Mönch mit dir spielt. Ich denke, dass mehr Spiele einen Mönch enthalten sollten.

Die Regeln sind einfach. Wir haben einen Stapel von 87 Steinen, die in ein Feld mit 96 Zellen platziert werden sollen. Die ersten 6 Steine werden zu Beginn des Spiels zufällig platziert. Oben rechts wird angezeigt, welcher Stein gerade platziert werden soll. Die Platzierung ist immer neben einem anderen Stein, der den Regeln entspricht, dass entweder die Farbe oder das Symbol übereinstimmt. Das Ziel des Spiels ist es, so viele wie möglich zu platzieren. Weitere Punkte gibt es für „4ways“, das ist wenn man um einen Stein vier weitere legt, jeweils zwei in derselben Farbe und zwei mit demselben Symbol. Das Spiel endet, wenn keine Steine mehr im Stapel sind oder man keine Steine mehr legen kann. Für detailliertere Regeln, lest einfach, was der Mönch sagt und spielt ein paar Spiele, dann wird es dir einfallen.

Die Grafik ist schön, detailliert und wir können aus mehreren verschiedenen Steinsymbolen und Hintergründen wählen. Es gibt auch AY Musik und Geräusche für Signaltöne. Die Steuerung ist einfach und insgesamt die Spielbarkeit gut.

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/35329/ZX-Spectrum/The_Legacy_of_the_White_Crane

»THE LOST TREASURES OF TULUM«
© 2020 VON RETROWORKS (SEJUAN, WYZ, UTOPIAN)

Antike Schätze, Mythen und Legenden. Es gibt nicht allzu viele Menschen, die solche Dinge nicht mögen. Unser Held konnte auch nicht widerstehen und das hat dazu geführt, dass er in einem alten Tempel eingeschlossen ist. Es gibt einen Ausweg durch die Dunkelheit, aber die Schätze werden ihm den Weg weisen. Wirst du ihm helfen?

Ein weiterer Plattformer mit einer Besonderheit. In einem alten Tempel ohne Licht zu sein, ist keine gute Sache. Erinnerst du dich an **SWITCHBLADE**? Ein Spiel, das dir nur den Teil des Bildschirms zeigt, der dich umgibt, und du musst dich bewegen, um den ganzen Ort zu sehen. Der Tempel von Tulum ist ähnlich.

Jeder Level enthält einen Raum. Wir beginnen auf einem schwarzen Bildschirm und müssen uns in dem Raum bewegen, um alle Schätze zu finden (es sind vier in jedem Level). Dadurch öffnen sich die Türen und wir können zum nächsten Level weitergehen und schließlich den Tempel verlassen. Unser Held kann springen, klettern, rennen und kämpfen. Wir sind nicht allein, denn es gibt Wächter des Tempels. Sie tauchen aus der Dunkelheit auf und wenn wir außer Reichweite sind, tauchen sie wieder in die Dunkelheit ab. Aber sobald wir den ganzen Bildschirm aufdecken, können sie nicht mehr verschwinden und wir müssen zum Ausgang laufen. Und Vorsicht, die Feinde können auch springen und klettern! Es kann die Situation eintreten, dass alle Lichter verschwinden. Dann müssen wir zurück zu der Flamme gehen, die sich an der Stelle befindet, an der wir das Level begonnen haben, um unsere Kerze anzuzünden, damit wir den Raum wieder sehen können. Goldtruhen können geöffnet werden, indem man sie angreift,

manchmal ist das sogar nötig, da sie den Weg zum Schatz versperren.

Die Grafik ist wirklich schön und detailliert, die Animationen sind flüssig und die Steuerung sehr genau. Das Spiel ist nicht das einfachste überhaupt, aber zusammen mit einer schönen AY-Musik hat es eine gute Atmosphäre und bringt etwas Neues in den Stapel der modernen Plattformern. Ich empfehle dringend, einige Zeit mit diesem Spiel zu verbringen!

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/35776/ZX-Spectrum/The_Lost_Treasures_of_Tulum

»DIZZY VIII - WONDERFULL DIZZY«
©2020 THE OLIVER TWINS

Ich weiß gar nicht, wie ich anfangen soll. ***Dizzy*** ist zurück. Und nicht in einem Remake oder einem Fan Spiel. ***Dizzy*** ist zurück in seinem neuen, kollosalen Abenteuer und das von seinen ursprünglichen Autoren unterstützt. Dieses Mal haben sie mit vielen anderen Leuten aus der **Spectrum**-Szene zusammengearbeitet, um uns ein Deluxe-Spiel zu liefern, das Bestand haben wird. Also, lasst uns ***Dizzy*** bei seinen neuen Abenteuern helfen.

Während eines riesigen Sturms haben es ***Dizzy*** und ***Pogie*** im letzten Moment in ihre Hütte geschafft. Der Sturm war so stark, dass sie das Gefühl hatten, die ganze Hütte würde wegfliegen.. Und als er vorbei war, beeilten sie sich, hinauszugehen, um zu sehen, welchen Schaden er verursacht hat. Aber jetzt kommt die Überraschung - die Hütte ist wirklich geflogen und sie sind jetzt im Land des **Zauberers von OZ**. Das wäre ja noch ganz in Ordnung, aber als die Hütte landete, tat sie das direkt auf der bösen Hexe und ihre Stiefel sind das Einzige, was von ihr übrigblieb. Als ***Dizzy*** diese ausprobiert hat, taucht die Schwester der Hexe auf und nimmt Pogie mit. Der arme ***Dizzy*** muss nun quer durch das Land **OZ** reisen, um sie zu finden und ***Pogie*** zu retten.

Das Wichtigste zuerst; Das Spiel ist nur für **Spectrum 128k.** Aber es nutzt die Vorteile der Maschine aus und ist riesig. Außerdem gibt es eine Menge Texte zu lesen. Eine Karte wäre sehr praktisch, da einige Orte Abkürzungen enthalten oder von mehreren Seiten betreten werden können. Außerdem stapeln sich die zu sammelnden Objekte nach einer Weile des Spiels, so dass es offensichtlich wird, dass das Spiel ziemlich lange dauert, um es zu beenden. Achten Sie darauf, dass Sie Ihre Steuerungen richtig definieren, da man sie nicht mehr ändern kann, sobald man das Spiel startet.

Es gibt neue Dinge und zwar eine ganze Menge davon, also schauen wir uns diese an. Zunächst einmal die Grafik. Nach einem Schwarz-Weiß-Intro und dem Anfang wird man von den Farben überwältigt sein. Und der ***Dizzy*** selbst ist es auch, wie er zu Beginn des Spiels scherzt.

Die Grafik ist anders als bei der alten ***Dizzy***-Serie, manches erinnert an das ***Dizzy*** VII-Remake von vor ein paar Jahren. ***Dizzy*** hat jetzt einen Rucksack auf seinem Rücken, kann aber immer noch höchstens drei Gegenstände tragen. Er ist sehr schön animiert, einschließlich Aktionen wie das Betreten von Türen, das Erklimmen von Leitern und so weiter. Er setzt sich sogar hin, wenn man die Taste drückt. Da wir jetzt nicht mehr in Yolkfolk sind, trifft ***Dizzy*** auf neue Kreaturen. Und da wir jetzt im Land von OZ sind, werdet ihr sicher einige von ihnen wiedererkennen. Es gibt eine **böse Hexe**, eine **gute Hexe**, den **Zauberer von OZ** und viele andere Gestalten, mit denen ***Dizzy*** zu tun hat.

Während du dich bewegst, verändert sich die Umgebung. Das Dorf ist schön und hell, der Wald ist dunkel und gruselig. Das trägt viel zur Gesamtatmosphäre bei, die böse Hexe ist von Dunkelheit umgeben, die gute Hexe lebt in einem schönen und schicken Schloss.

Alles fühlt sich lebendig an, es gibt Kreaturen, die man meiden sollte (Schlangen und Fledermäuse zum Beispiel), es gibt Figuren, mit denen man reden kann. Wasser und Feuer sind beide schön animiert und alles, was sich bewegt, ist sehr flüssig.

Was ist mit den Rätseln? Nun, sie sind logisch. Wenn Sie ein Zahnrad finden und man danach zu dem Bildschirm, wo die Brücke fehlt, sieht man einen Hebel. Es ist naheliegend, es zu versuchen und ja, es funktioniert. Und weil es ***Dizzy*** ist, ist der Weg zur ist der Weg zur Rettung von ***Pogie*** lang, weil es einfach zu viele Figuren gibt, die Hilfe brauchen. Und ihre Bedürfnisse sind unterschiedlich.

Dizzy VIII hält sich an die Tradition der Action-Adventures, also musst du auch über die Plattformen springen, Feinden ausweichen und deine Züge planen. Da die Steuerung sehr genau ist, ist das keine große Sache und das Gameplay ist sehr flüssig. Wie ich bereits erwähnt habe, ist das größte Problem hier die Orientierung, denn die Spielkarte wird im Laufe des Spiels immer größer und größer.

Wenn du nur wenig Zeit für Spiele hast, aber trotzdem etwas Interessantes und Außergewöhnliches spielen möchtest etwas Interessantes und Außergewöhnliches spielen willst, ist ***Dizzy VIII*** die richtige Wahl. Du wirst nicht enttäuscht sein und viel Spaß beim Spielen haben!

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36568/ZX-Spectrum/Wonderful_Dizzy

»DUNGEONS OF GOMILANDIA«
©2020 VON RETROWORKS

Vielleicht erinnerst du dich an ***GOMMY***, einen Helden aus der Vergangenheit. In ***GOMMY, MEDIEVAL DEFENDER*** kämpfte er gegen feindliche Horden, die versuchten, in eine Burg einzudringen. Jetzt ist er zurück und braucht wieder unsere Hilfe. ***GOMMY*** wurde in einen Kerker geworfen, aus dem niemand je wieder herauskam. Es ist nicht voller Monstern und Magie, sondern sein aktueller Feind ist die Zeit. Sie tickt wirklich schnell!

Das Spiel ist ein schnelles Logikpuzzle. Der Kerker enthält 55 Räume, die wir durchqueren müssen, um das Ende zu erreichen. In jedem Raum müssen wir einen Schlüssel finden, der die Tür öffnet und dann den Raum verlässt. Das klingt recht einfach, aber wie immer gibt es einige Hindernisse. Das erste ist die Bewegung. Wir können einen Stein hoch und runter gehen. Wir können zwei Ziegelsteine fallen, aber nicht hochgehen. Wenn wir mehr als zwei Ziegelsteine herunterfallen, verlieren wir ein Leben. Wir können ein Ziegelstein nehmen, auf dem wir stehen, und das senkt unsere Höhe, wir können ihn auch auf den Platz legen, auf dem wir stehen, und das erhöht unsere Höhe. In späteren Levels werden wir auch verschiedene Steine bekommen, manche verschwinden langsam unter unseren Füßen, einige sind mit Pfeilen markiert. Diese können benutzt werden, um sich zu bewegen und sind sehr praktisch. Aber ihr müsst vorsichtig sein, denn sobald ihr sie anhaltet ohne die Möglichkeit, von ihnen abzusteigen, verliert man ein Leben. Es gibt auch einige "Sprung"-Bausteine, mit denen man größere Höhen erreichen kann.

Die Grafik ist schön und farbenfroh. Die Bewegung erfolgt mit Attributen, also sehr vorsichtig sein, da ***GOMMY*** mehr Schritte machen kann als gewünscht und das führt oft zu seinem Tod. Die Geräusche sind für den **SPECTRUM 48K** und es gibt reichlich davon. Zu Beginn jedes Levels bekommen wir ein Passwort, damit wir nicht das ganze Spiel das ganze Spiel auf einmal spielen müssen. Das macht es zu einem tollen Spiel für ein paar kurze Durchläufe, wenn man in der Stimmung ist, etwas Schnelles zu spielen, bei dem man sein Gehirn benutzen muss. Das Spiel enthält auch einen entspannten Modus, in dem wir kein Zeitlimit haben. Aber wir werden auch keine Punkte nach Abschluss der Levels erhalten. Ich empfehle dieses Spiel, es wird dich nicht enttäuschen.

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/23487/ZX-Spectrum/Gommy_Defensor_Medieval

»HELL YEAH!<<
© 2020 VON ANDY PRECIOUS

Der Name ist seltsam, aber wenn man bedenkt, dass es Andys Debüt auf dem **SPECTRUM** ist, macht es Sinn. Ich habe es ähnlich genannt, nachdem ich es zum ersten Mal gesehen habe. So, sehen wir es uns genauer an.

Wie immer liegt es an dir, die Welt zu retten. Dazu hast du eine Waffe und musst über das feindliche Territorium laufen. Während du Horden von Soldaten, Dämonen und Monstern eliminierst, sammelst du auch Energiekisten, Erste-Hilfe-Pakete und Bonuswaffen. Und du feuerst weiter. Habe ich schon erwähnt, dass du rennst? Denn das Spiel ist schnell!

Die Grafik ist groß, farbenfroh und sehr schnell. Es gibt minimale Farbzusammenstöße und das alles sieht einfach toll aus. Weil sich alles mit 8 Pixeln bewegt bedeutet das Geschwindigkeit. Auf normalem Terrain ist das kein Problem, aber sobald man anhalten muss um Sprünge zu planen, wird es ein bisschen schwieriger. Es besteht keine Notwendigkeit für pixelgenaue (Attribut-)Sprünge, aber da die Feinde ständig auf dem Bildschirm auftauchen, ist die Action schnell und es ist nicht einfach, sich auf andere Dinge als Schießen zu konzentrieren.

Ich muss sagen, dass das Spiel wirklich frustrierend ist. Ich habe zwei Tage gebraucht, um mich hineinzukommen. Es braucht viel Übung, um auf die Plattformen, Bäume und andere Hindernisse zu springen. Zum Glück bekommen wir im Laufe des Spiels immer stärkere Waffen. Sie haben begrenzte Munition, sind aber eine gute Hilfe, besonders bei größeren Gegnern.

Die Geräusche sind für **SPECTRUM 48K**, aber sie sind ziemlich gut. Die Gesamtatmosphäre des Spiels ist gut und als hektischer Action-Shooter kann ich es nur empfehlen.

Probiert es aus, wenn ihr schon genug von Rätselspielen habt. Oh, und benutzt keine Kassette, das Spiel lädt die Levels nach.

Herunterladen unter:

https://spectrumcomputing.co.uk/entry/36497/ZX-Spectrum/HELL_YEAH

sintech

www.sintech-shop.de

Ersatzteile, Zubehör und Reparaturen
für Mobiltelefone, Spielekonsolen und mehr

SINTECH GmbH
Kurmainzstr. 14 - 74389 Cleebronn

Bei Online-Bestellungen zuzüglich Versandkosten. Es gelten unsere AGB's unter https://www.sintech-shop.de/gtc

sintech DEUTSCHLAND sintech CZECH REPUBLIC sintech UNITED KINGDOM

→ WANN FINDET DER EVENT STATT?

SAMSTAG, 30. APRIL 2022

→ UND WO?

BESCOT (BANK'S) STADIUM , WALLSALL

→ ANREISE:

MIT DEM AUTO ÜBER DEN M6 MOTORWAY,
UND ÖFFENTLICHE VERKEHRSMITTEL

→ ÖFFNUNGSZEITEN:

DIE SHOW BEGINNT UM 9:00 UHR MORGENS
UND ENDET AM ABEND UM 22:30 UHR

→ WAS KOSTET MICH DER SPASS?

SPECTRUM NEXT GEWINNSPIEL-TICKET: 5£
TAGESKARTE EIN ERWACHSENER: 20£
TAGESKARTE ZWEI ERWACHSENE: 35£
TAGESKARTE KINDER VON 5 BIS 16 JAHREN: 15£
FAMILIEN-TAGESKARTE: 50£
PARTY-TICKET: 10£

→ WAS WIRD GEBOTEN?

GÄSTE AUS 40 JAHREN ZX SPECTRUM:
TIM GILBERTS, THE OLIVER-TWINS, MELVIT DINC, MARK HIBBETT...
ZX FAIR VERKAUFSSTÄNDE,
ZX SPECTRUM NEXT – VERLOSUNG,
UND VIELES MEHR...

→WEITERE INFOS UNTER

HTTPS://SPECTRUMEVENTS.WEBSITE/

ODER

THOMAS EBERLE
THOMAS.EBERLE@SINTECH-SHOP.DE